Exploring Behaviour Patterns with Self-Organizing Map for Personalised Mental Stress Detection

Master's Thesis in Data Science
Jaakko Tervonen
Research Unit of Mathematical Sciences
University of Oulu
Spring 2019
Advisors: Jani Mäntyjärvi and Mikko Sillanpää

Abstract

Stress is an important health problem and the cause for many illnesses and working days lost. It is often measured with different questionnaires that capture only the current stress levels and may come in too late for early prevention. They are also prone to subjective inaccuracies since the feeling of stress, and the physiological response to it, have been found to be individual. Real-time stress detectors, trained on biosignals like heart rate variability, exist but majority of them employ supervised learning which requires collecting a large amount of labelled data from each system user. Commonly, they are tested in situations where the stress response is deliberately induced (e.g. laboratory). Thus they may not generalise to real-life conditions where more general behavioural data could be used.

In this study the issues with labelling and individuality are addressed by fitting unsupervised stress detection models at several personalisation levels. The method explored, the Self-Organizing Map, is combined with different clustering algorithms to find personal, semi-personal and general behaviour patterns that are converted to stress predictions. Laboratory biosignal-data are used for method validation. To provide an always-on type stress detection, real-life behavioural data consisting of biosignals and smartphone data are experimented on.

The results show that personalisation does improve the predictions. The best classification performance for the laboratory data was found with the fully personalised model (F1-score 0.89 vs. 0.45 with the general model) but for the real-life data there was no big difference between fully personal (F1-score 0.57) and general model as long as the behaviour patterns were mapped to stress individually (F1-score 0.60).

While the scores also validate the feasibility of SOM for mental stress detection, further research is needed to determine the most suitable and practical level of personalisation and an unambiguous mapping between behaviour patterns and stress.

Tiivistelmä

Stressi on merkittävä terveysongelma ja syynä useisiin sairauksiin sekä työpoissaoloihin. Sitä mitataan usein erilaisilla kyselyillä, jotka kuvaavat vain hetkellistä stressitasoa ja joihin voidaan vastata liian myöhään ennaltaehkäisyn kannalta. Kyselyt ovat myös alttiita subjektiivisille epätarkkuuksille, koska stressintunteen, ja stressinaikaisten fysiologisten reaktioiden, on havaittu olevan yksilöllisiä. Reaaliaikaisia, biosignaalien kuten sykevälivaihtelun analyysiin perustuvia, stressintunnistimia on olemassa, mutta pääosin ne käyttävät ohjatun oppimisen menetelmiä, mikä vaatii jokaiselta järjestelmän käyttäjältä suuren stressintunteella merkityn aineiston. Stressintunnistimia myös usein testataan tilanteissa, joissa stressi on tahallisesti aiheutettua (esimerkiksi laboratoriossa). Siten ne eivät yleisty tosielämän tarpeisiin, jolloin voidaan käyttää yleisempää käyttäytymistä kuvaavaa aineistoa.

Tässä tutkimuksessa vastataan datan merkintäongelmaan sekä yksilöllisyyden huomioimiseen käyttäen ohjaamattoman oppimisen stressintunnistusmalleja eri yksilöimisen tasoilla. Käytetty menetelmä, itseorganisoituva kartta, yhdistetään eri ryhmittelyalgoritmeihin tavoitteena löytää henkilökohtaiset, osin henkilökohtaiset sekä yleiset käyttäytymismallit, jotka muunnetaan stressiennusteiksi. Menetelmän sopivuuden vahvistamiseksi käytetään laboratoriossa kerättyä biosignaalidataa. Menetelmää sovelletaan myös tosielämän stressintunnistukseen biosignaaleista ja älypuhelimen käyttödatasta koostuvalla käyttäytymisaineistolla.

Tulokset osoittavat, että yksilöiminen parantaa ennustetarkkuutta. Laboratorioaineistolla paras luokittelutarkkuus löydettiin täysin yksilöllisellä mallilla (F1-pistemäärä 0.89, kun yleisellä 0.45). Tosielämän aineistolla täysin yksilöllisen (F1-pistemäärä 0.57) ja yleisen mallin, jossa käyttäytymismallien ja stressin välinen kuvaus määrättiin yksilöidysti (F1-pistemäärä 0.60), välinen ero ei ollut suuri.

Vaikka tulokset vahvistavatkin itseorganisoituvan kartan sopivuuden psyykkisen stressin tunnistamisessa, lisätutkimusta tarvitaan määräämään soveltuvin ja käytännöllisin yksilöimisen taso sekä yksikäsitteinen kuvaus käyttäytymismallien ja stressin välille.

Acknowledgements

This thesis was done in the Data-Driven Life group at VTT, The Technical Research Centre of Finland.

First, I would sincerely like to thank my supervisor at VTT, Jani Mäntyjärvi, Dr. Tech., for introducing me to behavioural analysis and for his guidance and suggestions throughout the process. I thank my supervisor at the university, Professor Mikko Sillanpää, for the many discussions during the autumn when we still didn't even have an explicit subject for the thesis, and for the comments on improving the text.

I thank all my co-workers at the office and I thank all the people at VTT and FIOH, The Finnish Institute of Occupational Health, who took part in collecting and preprocessing the real-life data. For funding the project I thank Business Finland, VTT and FIOH.

Finally, I am grateful to my friends and family, and especially Essi, for taking my mind off from work whenever I needed it.

Jaakko Tervonen
Oulu, 3.4.2019

Abbreviations

ANOVA	Analysis of Variance, a statistical method
ARI	Adjusted Rand Index, a clustering metric
BMU	Best Matching Unit, the SOM neuron most similar to a data point
BVP	Blood Volume Pulse, the volume of blood passing through the tissues at certain area with each heart beat
ECG	Electrocardiogram, electrical activity of the heart
EDA	Electrodermal Activity, skin conductance
e.g.	exempli gratia, for example
EM	Expectation Maximization, an iterative method to find estimates for model parameters
EMG	Electromyogram, electrical activity of skeletal muscles
EU	European Union
EU-OSHA	European Agency for Safety and Health at Work
GMM	Gaussian Mixture Model, a clustering method
GPS	Global Positioning System
HDBSCAN*	Hierarchical Density-Based Spatial Clustering of Applications with Noise, a clustering method
HMM	Hidden Markov Model, an unsupervised learning technique
HR	Heart Rate
HRV	Heart Rate Variability, variation in the time between heartbeats
IBI	Interbeat Interval, the time between heartbeats
i.e.	id est, that is
LODO	Leave-One-Day-Out, a cross-validation procedure
LOSO	Leave-One-Subject-Out, a cross-validation procedure
MCMC	Markov Chain Monte Carlo, a class of sampling methods
PPG	Photoplethysmography, a technique to detect blood volume changes
RMSSD	Root Mean Square of Successive IBI Differences, an HRV measure
SOM	Self-Organizing Map, an unsupervised artificial neural network
SVM	Support Vector Machine, a classification method
WESAD	Wearable Stress and Affect Detection, an open-source dataset

Contents

Abstract

Tiivistelmä

Acknowledgements

Abbreviations

1 Introduction **6**

1.1 Defining and Detecting Stress 6

1.2 Challenges in Stress Detection 7

1.3 Related Work 8

1.4 Scope and Structure of the Thesis 10

2 Methods **12**

2.1 Clustering 13

2.1.1 Mixture Models 13

2.1.2 Density-Based Clustering 19

2.1.3 Evaluating Clustering Performance 21

2.2 Self-Organizing Map 25

2.2.1 Training Algorithm 27

2.2.2 Visualisation & Clustering 30

2.2.3 Setting Hyperparameters 32

2.3 Models and Personalisation 33

3 Experiments **37**

3.1 Laboratory Data 37

3.1.1 Data Preprocessing 39

3.1.2 SOM and Clustering 42

3.1.3 Results and Discussion 45

3.2 Real-Life Data 52

3.2.1 Data Description and Quality 52

3.2.2 Assessing Stress 57

3.2.3 Data Preprocessing 60

3.2.4 Results 62

4 Discussion and Future Work **76**

References **79**

1 Introduction

Psychosocial stress is a major problem in today's society. According to European Agency for Safety and Health at Work (EU-OSHA, as short), half of the European workers say that stress is common in their workplace, being the second most commonly reported health issue (EU-OSHA, 2013). They estimated that it is the reason for more than half of all working days lost, and the costs of work-related stress have been found to run into dozens of billions of euros per year in the EU (Hassard et al., 2014).

The health problems associated with prolonged, long-term stress include increased chance of mental health problems (depression), cardiovascular diseases, musculoskeletal disorders and diabetes (Hassard et al., 2014). Work-induced stress may also increase sickness leaves and absenteeism, resulting in increased company overhead and public health care costs (Alberdi et al., 2016; Hassard et al., 2014). Therefore, it is important to detect and treat stress as early as possible to minimise the risks and related costs.

1.1 Defining and Detecting Stress

Originally stress is defined as "the non-specific response of the body to any demand for change" (Selye, 1956). More recent definitions have taken into account one's ability to deal with the changes, and stress is said to occur when there is an imbalance between external forces (e.g. demands of work) and individuals ability to cope with them (Lazarus, 1993; EU-OSHA, 2013). While some amount of stress is normal and can even have positive effects on performance (*eustress*, introduced by (Selye, 1956)), it is the negative stress, *distress*, that is usually understood as stress.

Stress can be further divided into *acute*, *episodic acute* and *chronic* stress (Bakker et al., 2011). Acute stress is the type of stress most people experience in everyday life, caused by some short-term stress factor, and is not considered harmful. Stress turns to episodic as the frequency of acute stress increases and physiological symptoms may start to appear (Bakker et al., 2011). The most harmful

type of stress is chronic stress which takes place when stress factors are persistent (Bakker et al., 2011).

Stress detection is often done by the means of self-report questionnaires, like the Perceived Stress Scale (Alberdi et al., 2016; Sharma and Gedeon, 2012). These methods are considered reliable but they only reflect responses at spot-checks, offering information on the current level of stress and not about the causes or evolution of stress levels (Alberdi et al., 2016). Furthermore, the tests are usually taken only after stress is too severe for early prevention (Alberdi et al., 2016), which is why stress detection research has focused on developing means for real-time stress monitoring. As stress can be detected from a variety of biosignals like heart rate variability, respiration or electrodermal activity (Sharma and Gedeon, 2012), in recent years there has been a large interest in developing an automated stress recognition system usually based on biosignal or smartphone usage data, or both, e.g. (Huysmans et al., 2018; Sano et al., 2018; Schmidt et al., 2018; Smets et al., 2018; Taylor et al., 2017; Vildjiounaite et al., 2017, 2018).

1.2 Challenges in Stress Detection

As noted by (Alberdi et al., 2016), a majority of stress recognition methods proposed experiment on data collected in a laboratory setting, making them hard to generalise to real-life context. In contrast to the controlled conditions of the laboratory, real-life data are much more abundant, diverse and disarranged, and we do not know the circumstances that led to any specific situation (background data) to arise. This means that we have big data with necessarily no ground truth on its underlying structure. That is a problem because the most usual methods in stress detection like Support Vector Machines (SVM), k-Nearest Neighbours, Linear Discriminant Analysis and Artificial Neural Networks are all supervised learning methods, requiring labels to guide the learning process (Sharma and Gedeon, 2012; Alberdi et al., 2016). Obtaining correct and accurate stress labels in laboratory conditions is quite straightforward because the study protocol is designed to produce the wanted reaction. Questionnaires or self-reports are often used for further ground truth. In a real-life setting the situation is not so simple and obtaining

enough accurate and truthful labels can be challenging.

An addition to the labelling problem, the feeling of stress is subjective and it has been observed that some people may be more inclined to report stress than others. An often used personality questionnaire is the Big Five personality trait test that evaluates the person's levels of openness, conscientiousness, extraversion, agreeableness and neuroticism. In an exploratory study by (Ervasti et al., 2019), it was found that high level of neuroticism was associated with higher level of self-reported stress, and higher levels of extraversion, agreeableness, and conscientiousness were associated with lower self-reported stress. Further, (Vildjiounaite et al., 2017) found that their stress detection system's recognition rate of high stress negatively correlated with the conscientiousness and positively correlated with the openness score. This may indicate that those with high conscientiousness score have not observed or reported stress even if there had been a stress reaction and those with high openness score have more freely admitted to having stress. The system had around 75% accuracy of correctly recognizing a reported stress reaction.

If people's tendency to report and feel stress differs, so do their reactions to stress factors. As proved by (Healey, 2000), inter-subject physiological responses to stress can vary significantly. It is also clear that smartphone usage patterns or human behaviour patterns (routines) in general are different between individuals.

Both issues regarding labelling and individual response suggest that some level of (or full) personalisation of models is needed to capture the subject-to-subject deviation. Several options to overcome the challenges have been proposed. Next, we go through some of them and report the found stress detection scores.

1.3 Related Work

To answer the labelling problem in a real-life situation, the most usual method is questionnaires several times a day. However, it has been found that people tend to answer them rarely. (Kusserow et al., 2013) used a diary of daily activities and mood-state questionnaires to monitor the stress-arousal phase of the participants. They observed that most questionnaires were filled in randomly throughout the

day and could not be related to estimated stress-arousal phases. (Adams et al., 2014) collected data from seven participants over a ten day period, with self-report notifications emerging on their smartphones every half an hour. Over the ten-day period, the participants answered 28% of the self-reports on average and many of the responses were delayed due to technical difficulties or occupation of participants. As a larger scale example, (Smets et al., 2018) collected five days of data from 1002 subjects with twelve pop-up questionnaires per day. For them, 920 subjects answered at least one questionnaire, with an average compliance rate at 42%. The first two did not attempt stress detection and the last obtained an F1-score of 0.43.

Another solution to the labelling issue is the use of unsupervised methods that require no labels to train the model. The problem with this approach is result validation which in turn requires labels. To date, unsupervised methods have rarely been used. Out of the 44 papers summarised in the review by (Alberdi et al., 2016), three used unsupervised methods, namely the Hidden Markov Model (HMM), but they did not comment whether personalisation was used. Since then, (Vildjiounaite et al., 2017) trained general, semi-personal and fully personal HMM models with real-life biosignal and smartphone usage data and found that the stress detection rate with the fully personal model was the highest (75%) in all the tests run. In another paper, they compared semi-personal to fully personal models using real-life smartphone data only and found that the personal version worked better with a detection accuracy of up to 81% (Vildjiounaite et al., 2018). As another example of an unsupervised method, (Huysmans et al., 2018) managed to get a stress detection accuracy of 79% with a semi-personal Self-Organizing Map in a laboratory study - they did not report scores for personal or general models.

Personalisation has also been used successfully in supervised settings. (Shi et al., 2010) made a personalised version of SVM and found an increase of up to 6% in positive predicted value compared to the general model (highest positive predictive value 62% at 80% sensitivity). (Xu et al., 2015) clustered the subjects first with the K-Means Clustering algorithm, and then used cluster-wise (i.e. semi-personal) regression neural networks for stress detection to obtain an accuracy of

85% with the personalised model which is 12% higher than the general model. In a study by (Smets et al., 2016), several supervised methods were compared with both general and personalised models, achieving the best detection rate of 84.6% with personalised dynamic Bayesian Networks; however, on average personalised models did not work as well as generalised ones in their study. In a wider context of tomorrow's mood prediction, (Taylor et al., 2017) got an average increase of 16.4% in the accuracy of stress prediction task by the use of semi-personalised models compared to the general version, with the highest accuracy of 81.5%. The first three were laboratory studies, while the last used real-life data.

1.4 Scope and Structure of the Thesis

In this study, we explore the Self-Organizing Map combined with different clustering algorithms as a way to train stress detection models without the need for ground truth labels. To account for the individuality in feeling and reporting stress, we experiment on different levels of personalisation. To date, previous research has mostly focused on considering only one level of personalisation but we study its effects on multiple levels.

In addition, previous studies to stress detection with unsupervised methods have used data either from a laboratory experiment or real-life conditions but not both. The data collected in a laboratory are usually of high quality and sample-by-sample annotations are available, which is generally not the case for real-life data. Therefore, we attempt on bridging the gap between laboratory and real-life studies by validating our approach on laboratory data and then generalising it to the needs of real-life data. The effects of personalisation and different clustering options are investigated for both kinds of data.

With these aspects in mind, we are interested in finding the answers to the following questions:

1. Can stress be detected from continuous behavioural measurements and how do laboratory and real-life measurements differ?

2. How can the Self-Organizing Map be used to detect and extract behaviour

patterns? Can they be related to stress and how?

3. What is the effect of personalisation and is it needed?

In Chapter 2 we introduce our approach to unsupervised stress detection. We go through the mathematics of the method and present the Self-Organizing Map and three clustering algorithms we combine with it. We also introduce the ways we personalise the models. Chapter 3 is dedicated to applying the model to two datasets, one collected in a laboratory and the other in real-life conditions. Chapter 4 concludes this thesis by summarising the results and possible future research.

2 Methods

Statistical learning is the set of tools to understand and draw conclusions from data and it is often divided to *supervised* and *unsupervised* learning. In supervised learning, the goal is estimating a function mapping input data $\mathbf{X}$ to output data $\mathbf{y}$ when both are known and can be used for estimating the function. For example, in linear regression, we aim at finding such a function $\mathbf{f}$ that $\mathbf{y} = \mathbf{f}(\mathbf{X}) = \mathbf{X}\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is the vector of linear coefficients. In general this relationship can only be learned up to an error ε and an approximation of the function $\mathbf{f}$ is found by minimising the squared L2-norm $||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2$.

In unsupervised learning, the output data are not known. Some examples of the goals of the unsupervised learning process are finding a lower dimensional representation of the input data (dimensionality reduction), looking for groups within the data (clustering) or identifying abnormalities or outliers (anomaly detection). Of course, as with supervised learning, unsupervised learning can often be viewed as learning a function of input data. For example, any dimensionality reduction method attempts to find a function $\mathbf{g} : \mathbb{R}^{n \times m} \to \mathbb{R}^{n \times m'}, m' < m$, that gives a re-representation $\mathbf{X}' = \mathbf{g}(\mathbf{X})$ of input data minimising the amount of information lost.

In this study, we are interested in identifying stress reactions from high dimensional sensor data. Finding different reactions is basically the problem of finding groups within the data, with one or more groups corresponding to stress. As we apply unsupervised methods, seeking such groups requires cluster analysis. To allow for visualisation of the data and clusters found, a low dimensional representation of the data is needed first.

The Self-Organizing Map (SOM) (Kohonen, 1982) is an artificial neural network that is good at providing such low dimensional representation and visualisation of high dimensional data. As seen in (Vesanto and Alhoniemi, 2000), SOM can be combined with clustering to obtain a two-dimensional view of the groups in data. In addition, (Vesanto and Alhoniemi, 2000) showed that using SOM at the first stage instead of direct clustering reduces computation time without major loss of

performance.

In this chapter, these methods are thoroughly presented. In Section 2.1, we go through the clustering methods applied in the analysis and take a look at metrics to estimate the goodness of clustering. Section 2.2 is dedicated to SOM. After an intuitive look, we will cover the training algorithm, visualisation options and clustering the SOM prototypes. Finally, we will introduce the models and personalisation options used in the analysis.

2.1 Clustering

By definition, clustering is the process of dividing the input data $\mathbf{X}$ to c groups, called clusters. If viewed as a function learning problem, cluster analysis attempts to find a function

$$\mathbf{f} : \mathbb{R}^{n \times m} \to \mathbb{N}_c, \quad \text{where } \mathbb{N}_c = \{i \in \mathbb{N} : i < c\}, \tag{2.1}$$

maximising the within-cluster similarity (*cohesion*) and inter-cluster dissimilarity (*separation*).

Many clustering methods always find a given number, and exactly the given number, of clusters in the data, even if there is no real group structure in the data. In two- or three-dimensional case, it is easy to draw scatter plots of the data and assess the clusterability of the data. This is demonstrated in Figure 1. However, in p dimensional case, $p(p-1)/2$ two-dimensional scatter plots are needed and visual inspection is not viable. Later in Section 2.2.2, we will see how the SOM U-matrix allows to visually assess the clusterability of the data.

2.1.1 Mixture Models

Suppose we have a dataset $\mathbf{X} \in \mathbb{R}^{n \times m}$, where $\mathbf{X}$ consists of m-dimensional vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Our goal is to partition the data to c groups, each having a probability distribution with density function f_k. Now, the probability density function for the samples in $\mathbf{x} \in \mathbf{X}$ is given by a mixture density

$$f(\mathbf{x}) = \sum_{k=1}^{c} w_k f_k(\mathbf{x}|\theta_k), \tag{2.2}$$

Figure 1: Clustering two datasets with K-Means Clustering, with K=3. On the left panel the correct class labels are depicted with markers and clustering results with colors. On the right all the points come from the same distribution but clusters are found anyway. The silhouette scores (Eq. 2.22) for the two outcomes are 0.64 and 0.33, respectively.

where w_k are the mixture weights and θ_k are the parameters related to density f_k.

In this section, two clustering algorithms based on mixture models are presented, namely the *K-Means Clustering* and *Gaussian Mixture Models* (GMM). We progress by first deriving the K-Means algorithm, then presenting GMMs and finally, we will see how K-Means can be viewed as a special case of GMMs.

K-Means

K-Means summarises the notions of cohesion and separation by trying to assign each data point $\mathbf{x}_j$ to a cluster k in a way that its distance to elements in other

clusters is higher than its distance to the elements in the same cluster. If we let $\boldsymbol{\mu}_k, k = 1, \ldots, c$, denote a prototype of cluster k, our goal now is to minimise the sum of squares of the distances of each data point to its closest vector $\boldsymbol{\mu}_k$ (Bishop, 2006, p. 424). The K in K-Means refers to the number of clusters but here we will retain our notation c.

For convenience, let us define a binary variable $r_{jk} \in \{0, 1\}$ indicating the cluster k the data point $\mathbf{x}_j$ is assigned to. Now, an objective function J representing the sum of squares of each data point's distance to its prototype $\boldsymbol{\mu}_k$ is given by (Bishop, 2006, p. 424)

$$J = \sum_{j=1}^{n} \sum_{k=1}^{c} r_{jk} ||\mathbf{x}_j - \boldsymbol{\mu}_k||^2, \tag{2.3}$$

where $|| \cdot ||$ is the Euclidean distance. More generally, the Euclidean distance can be replaced with any well-defined metric.

At this stage we already know that we do not necessarily find a global minimum for J. Since $r_{jk} \in \{0, 1\}$ for all $j = 1, \ldots, n$ and $k = 1, \ldots, c$, the range of r_{jk} is non-convex. Therefore, the domain of J is non-convex and J is not a convex function, thus not guaranteeing a global solution.

The minimisation of (2.3) is done by the Expectation-Maximisation, or the EM-algorithm, the general version of which was originally presented by (Dempster et al., 1977). At E-step, we minimise J with respect to r_{jk}. This is done easily by letting $r_{jk} = 1$ for whichever value of k that gives the minimum value of $||\mathbf{x}_j - \boldsymbol{\mu}_k||^2$. At M-step, we optimise with respect to $\boldsymbol{\mu}_k$. Now, J is a quadratic function of $\boldsymbol{\mu}_k$, and

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} J = 2 \sum_{j=1}^{n} r_{jk} (\mathbf{x}_j - \boldsymbol{\mu}_k). \tag{2.4}$$

Setting the derivative to zero and solving for $\boldsymbol{\mu}_k$ yields

$$\widehat{\boldsymbol{\mu}}_k = \frac{\sum_{j=1}^{n} r_{jk} \mathbf{x}_j}{\sum_{j=1}^{n} r_{jk}}. \tag{2.5}$$

The denominator equals the number of elements in cluster k, and so $\widehat{\boldsymbol{\mu}}_k$ is equal to the mean of all the points $\mathbf{x}_j$ in cluster k. The process of assigning points to clusters and re-computing the cluster means is repeated until convergence. (Bishop, 2006, p. 425)

K-Means is one of the most used clustering algorithms, probably due to its simplicity and easy implementation. The main drawback is that one must determine the number of clusters beforehand. Experimenting with different values for c must be done to conclude the most correct number of clusters. In addition, the algorithm assigns each point uniquely to one cluster, not accounting for the uncertainty related to the clustering process (Bishop, 2006, p. 428). As we will see later, K-Means also assumes that the cluster distributions are spherical, or that the cluster-wise covariance matrices $\mathbf{\Sigma}_k = \sigma^2\mathbf{I}$, where σ^2 is a variance parameter and $\mathbf{I}$ is the identity matrix.

Gaussian Mixtures

Let us rewrite (2.2) as

$$p(\mathbf{x}) = \sum_{k=1}^{c} \pi_k \phi_k(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k), \tag{2.6}$$

where $\phi_k(\cdot)$ is the cluster-wise density function of m-dimensional normal distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\mathbf{\Sigma}_k$, given by

$$\phi_k(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^m \det(\mathbf{\Sigma}_k)}} \exp\Big\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k^T)\mathbf{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\Big\}, \tag{2.7}$$

and π_k is the prior probability of a generic data point $\mathbf{x} \in \mathbf{X}$ of belonging to cluster k. Equation (2.6) is our starting point to fitting c Gaussian distributions with unknown mean vectors and covariance structures to our data $\mathbf{X}$. We follow the presentation in (Bishop, 2006, ch. 9.2) with some added details. As before, the subscript k always refers to a cluster and j to a data point.

Let $\mathbf{z}$ be a c-dimensional latent binary random variable in which a particular element $z_k = 1$ and $z_{k'} = 0$ for all $k, k' = 1, \ldots, c, k \neq k'$. We attempt on finding the maximum likelihood estimates for the parameters $\boldsymbol{\mu}, \mathbf{\Sigma}$ and $\boldsymbol{\pi}$ in (2.6). The maximization is done again with the EM-algorithm which is simplified by considering the joint distribution $p(\mathbf{x}, \mathbf{z})$ instead of the marginal distribution $p(\mathbf{x})$. By properties of conditional probability,

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}), \tag{2.8}$$

which shows that we will need to find $p(\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})$.

We specify the marginal distribution of $\mathbf{z}$ by the prior probabilities $\boldsymbol{\pi}$ by setting

$$p(z_k = 1) = \pi_k,$$

where $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$ to make sure that $p(\mathbf{z})$ determines a valid probability distribution. Because $\mathbf{z}$ is a latent binary variable, the equality above can be written in the form

$$p(\mathbf{z}) = \prod_{k=1}^{c} \pi_k^{z_k}. \tag{2.9}$$

The conditional distribution of $\mathbf{x}$ given a particular value for $\mathbf{z}$ is

$$p(\mathbf{x}|z_k = 1) = \phi_k(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

which, similarly to (2.9), can be written as

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{c} \phi_k(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{2.10}$$

Now, the marginal distribution of $\mathbf{x}$ is obtained by summing the joint distribution over all possible values of $\mathbf{z}$, giving

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{c} \pi_k \phi_k(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.11}$$

and we have the elements for the joint distribution (2.8). The likelihood for all the observations in $\mathbf{X}$ is now given by

$$L(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{X}) = \prod_{j=1}^{n} \sum_{k=1}^{c} \pi_k \phi_k(\mathbf{x}_j|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.12}$$

and its logarithm

$$\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{X}) = \sum_{j=1}^{n} \ln\Big\{ \sum_{k=1}^{c} \pi_k \phi_k(\mathbf{x}_j|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Big\}. \tag{2.13}$$

The process for maximizing this with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is relatively simple and only requires taking derivates and some algebraic manipulations. The constraint that prior probabilities $\boldsymbol{\pi}$ must sum up to one makes the case somewhat

more complex for maximising the likelihood with respect to $\boldsymbol{\pi}$, but it can be done by using a Lagrange multiplier. The whole maximisation process is presented in (Bishop, 2006, p. 435-436). Denoting

$$\begin{aligned}\gamma(z_k) &:= p(z_k = 1|\mathbf{x}) \\ &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{k'=1}^{c} p(z_{k'} = 1)p(\mathbf{x}|z_{k'} = 1)} = \frac{\pi_k \phi_k(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{c} \pi_{k'} \phi_{k'}(\mathbf{x}|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}\end{aligned} \tag{2.14}$$

and

$$N_k := \sum_{j=1}^{n} \gamma(z_{jk}) \tag{2.15}$$

lets us express the maximum likelihood estimates for $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and $\boldsymbol{\pi}$ as

$$\widehat{\boldsymbol{\mu}}_k = \frac{1}{N_k} \sum_{j=1}^{n} \gamma(z_{jk}) \mathbf{x}_j, \tag{2.16}$$

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{N_k} \sum_{j=1}^{n} \gamma(z_{jk})(\mathbf{x}_j - \boldsymbol{\mu}_k)(\mathbf{x}_j - \boldsymbol{\mu}_k)^T, \tag{2.17}$$

$$\widehat{\pi}_k = \frac{N_k}{n}. \tag{2.18}$$

The given maximum likelihood estimates do not provide a closed form solution, because they depend on $\gamma(z_k)$. Instead, an iterative EM-algorithm similar to K-Means algorithm can be used (Bishop, 2006, p. 436-439). A high-level version of this is given in Algorithm 1.

Algorithm 1 Gaussian Mixtures

1: Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixture coefficients π_k and evaluate the logarithmic likelihood (2.13).
2: **E-step.** Evaluate the responsibilities $\gamma(z_{jk})$ using (2.14) with the current parameter values.
3: **M-step.** Re-estimate the parameters using equations (2.16) -(2.18) with the current responsibilities.
4: Evaluate the logarithmic likelihood (2.13).
5: Repeat $2 - 4$ until convergence.

The quantity N_k is the effective number of points assigned to cluster k. The posterior probability $\gamma(z_k)$ of $z_k = 1$ after observing $\mathbf{x}$, obtained by The Bayes' Theorem, can be thought of as the responsibility that cluster k takes for explaining the observation. In other words, it is the probability that observation $\mathbf{x}$ belongs to cluster k. Because $0 \leq \gamma(z_k) \leq 1$, we see that GMM does not necessarily force the observation to one single cluster, but gives a probability of $\mathbf{x}$ belonging to each of the c clusters. This property is called *soft clustering*.

By comparing (2.16) to (2.5), we see the close resemblance of these two equations. Indeed, if we let $\mathbf{\Sigma}_k = \sigma^2\mathbf{I}$ for all $k = 1, \ldots, c$, where $\mathbf{I}$ is the identity matrix and σ^2 a variance parameter shared by all the components (clusters), then

$$\gamma(z_{jk}) \xrightarrow{\sigma^2 \to 0} r_{jk}$$

and (2.16) becomes (2.5). In addition, the expected complete data logarithmic likelihood becomes $-\frac{1}{2}J + C$ as $\sigma^2 \to 0$, where J is as in (2.3) and C is a constant. These results are justified in (Bishop, 2006, p. 443-444).

Thus K-Means differs from GMM in that it uses hard assignment of data points to clusters and it does not estimate the covariances of the clusters. They are both parametric, model-based clustering algorithms attempting to grasp the underlying probability distribution by the means of parameter estimation. When the data do not follow the model assumptions of these two methods and is more ill-behaved, as is often the case with real data, they tend to fail and find no meaningful clusters. While GMM allows for soft clustering, both of the methods cluster all the data. However, real data often come with noise and outliers that are rare and exceptional values and not necessarily part of any cluster. An algorithm able to handle these limitations is presented next.

2.1.2 Density-Based Clustering

In this Section, we turn our attention to a more recent technique called Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN*, Campello et al. (2015)), which is a non-parametric, density-based algorithm currently considered state-of-the-art clustering algorithm. As a high-level description,

it approximates the unknown probability distribution f the data are drawn from and divides the data into regions of high density (clusters) and low density (noise) in a hierarchical fashion. According to (McInnes and Healy, 2017), this is actually what clustering truly is, and K-Means and GMM are simply partitioning methods.

Framing the algorithm in a conceptual level, following (Campello et al., 2015) and (McInnes and Healy, 2017), first we need an estimate of density. Dense areas are effectively regions in the distribution with high concentration of data points. The density estimate λ at the point $\mathbf{x} \in \mathbf{X}$ is the reciprocal of *the core distance* $d_{core-k}(\mathbf{x})$ with respect to some parameter $k \in \mathbb{N}$, where $d_{core-k}(\mathbf{x})$ is the distance from $\mathbf{x}$ to its k'th nearest neighbour (Campello et al., 2015). Next up we define *the mutual reachability distance* between $\mathbf{x}_j$ and $\mathbf{x}_{j'}$ with respect to k as

$$d_{mreach-k}(\mathbf{x}_j, \mathbf{x}_{j'}) = \max\{d_{core-k}(\mathbf{x}_j), d_{core-k}(\mathbf{x}_{j'}), d(\mathbf{x}_j, \mathbf{x}_{j'})\}, \tag{2.19}$$

where $d(\mathbf{x}_j, \mathbf{x}_{j'})$ is the original distance between the two samples (Campello et al., 2015).

Mutual reachability distance preserves the distance between dense points (with low core distance) but increases the distance between sparse points up until they are at least their core distance away from any other point.

Under this new metric, we apply *Hierarchical Clustering with Single Linkage* (e.g. (Duda et al., 2000, p. 550-556)) to find areas with high density (McInnes and Healy, 2017). This gives us a *cluster tree*, with all the observations in a single cluster at the root of the tree with some of them dropping out from the cluster as we move (in terms of density) towards the tree leaves, which is the level at which each observation forms a cluster of its own.

We could cut this tree at a density level λ, given as a parameter to the algorithm, but choosing this parameter is troublesome. Instead, (Campello et al., 2015) introduce a parameter m_{cl} called *minimum cluster size*. As we move through the cluster tree starting from the root, some of the points, often only one or two, fall off from a cluster at various levels of λ. At cluster split, any child cluster having fewer than m_{cl} points are labelled as points "falling out of the parent cluster" at level λ. If only one cluster contains more than m_{cl} samples, we consider it a continuation of the parent, thus persisting the parent cluster's label. If more than one

child cluster contains more than m_{cl} points, we consider it a true split. This way we get a tree with a smaller number of clusters. (McInnes and Healy, 2017).

All that remains is extracting cluster labels for each data point which is done by maximising total *persistence*. For cluster C_i, define $\lambda_{max,C_i}(\mathbf{x}_j)$ to be the λ value at which the point $\mathbf{x}_j$ falls out of the cluster C_i either as an individual point or as a cluster split. Similarly, let $\lambda_{min,C_i}(\mathbf{x}_j)$ be the minimum value λ for which $\mathbf{x}_j$ is in C_i. Then the stability of the cluster C_i is given by (McInnes and Healy, 2017)

$$\sigma(C_i) = \sum_{\mathbf{x}_j \in C_i} (\lambda_{max,C_i}(\mathbf{x}_j) - \lambda_{min,C_i}(\mathbf{x}_j)). \tag{2.20}$$

The optimal clustering is obtained as the solution to the following maximisation problem. If there are n clusters, we wish to select $I \subseteq \{1, 2, \ldots, n\}$ to maximise

$$\sum_{i \in I} \sigma(C_i) \tag{2.21}$$

subject to the constraint that $C_i \cap C_{i'} = \emptyset$ for all $i, i' \in I, i \neq i'$. This means that we want to maximise persistence over the chosen clusters subject to the constraint that clusters must not overlap (McInnes and Healy, 2017). The points that are not in any of the resulting clusters are labelled as noise.

As seen, HDBSCAN* relies on two parameters, k and m_{cl}. To further simplify the algorithm, (Campello et al., 2015) suggest choosing $k = m_{cl}$ and so the only parameter to choose is minimum cluster size. As noted by (McInnes and Healy, 2017), the algorithm as presented here is in its most compact and conceptual form. They also deliver statistically and topologically motivated descriptions of the procedure.

2.1.3 Evaluating Clustering Performance

To estimate the performance of a clustering algorithm, we want to estimate cohesion and separation. The two factors are summarised in the *silhouette score* (Rousseeuw, 1987). The silhouette coefficient for sample i is given by

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{2.22}$$

where $a(i)$ is the mean distance from sample i to all other samples in the same cluster, and $b(i)$ is the mean distance of sample i to all the samples in the nearest cluster. Silhouette score is the mean of silhouette coefficients across all the samples $i = 1, \ldots, n$.

It is easy to deduce that $-1 \leq s(i) \leq 1$. Clearly $s = 0$ when $a = b$. By the definition of distance, $a \geq 0$ and $b \geq 0$. If $a < b$, then

$$s = \frac{b - a}{b} = 1 - \frac{a}{b}, \tag{2.23}$$

which gives that $0 < s \leq 1$. If $a = 0$ (this is the case when cluster A contains only one point (Rousseeuw, 1987) or all the elements in cluster A are actually multiple realisations of the same instance), the equation above equals to 1. If $a > b$,

$$s = \frac{b - a}{a} = \frac{b}{a} - 1, \tag{2.24}$$

which gives that $-1 \leq s < 0$. Value -1 is reached when $b = 0$. Here we have dropped the argument i from s, a and b for the sake of clarity.

When a gets small values, the within-cluster distances are small which corresponds to high cohesion. Consequently, s gets positive values as long as $a < b$. Similarly, high values of b mean high inter-cluster distances and thus high separation. Now, s gets positive values as long as $a < b$. Therefore, positive values of s refer to high cohesion and separation and thus good clustering, and negative values refer to poorer clustering. The requirement $a < b$ is not limiting since we want a to get small values and b to get high values. The extreme values 1 and -1 of s are reached in the cases of highly dense and well-separated clustering, and incorrect clustering, respectively.

The silhouette score works well in the case when the correct underlying labels are not known. If they are, we can simply compare how well the cluster labels correspond to the correct labels. This is always the case in any classification task. In this context, the analogue for cluster labels is predicted labels. The basic metrics used in classification include *accuracy*, the amount of correctly classified samples, *the positive predictive value* (PPV), reliability of positive predictions, *sensitivity*, probability of detection, and the *F1-score*, the harmonic mean of PPV and sensitiv-

ity (Fawcett, 2006). Table 1 depicts the confusion matrix of a binary classification task.

Table 1: The confusion matrix for binary classification task.

True / Predicted	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Using notation from the confusion matrix above, these measures are given as

$$\text{Accuracy} : \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad \text{PPV}: \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{F1-score} : \frac{2 \cdot \text{PPV} \cdot \text{sensitivity}}{\text{PPV} + \text{sensitivity}}, \quad \text{Sensitivity}: \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Accuracy generalises straightforwardly to a multi-class situation. For the other measures, the weighted mean of one-class-versus-the-rest scores can be used. More classification metrics exist but these are the relevant ones for us.

Unfortunately, in a clustering setting simply calculating these metrics will not do. Consider the following: given four observations with cluster labels $0, 0, 1, 1$ and correct labels $1, 1, 0, 0$ corresponds to a perfect clustering up to permutation of labels. However, all the scores defined above would equal to zero in this simple example. The problem may be even more severe if the number of clusters found is different from the number of correct classes. A metric called *Adjusted Rand Index* (Hubert and Arabie, 1985) is able to handle both of these complications.

Let U denote the set of correct labels and V the set of cluster labels, with R and C, respectively, the number of distinct labels in each set and n_{ij} the number of objects having the labels u_i and v_j. The information on label overlap can be summarised in a contingency table, like in Table 2. Note that the total number of point pairs in the table is $\binom{n}{2}$.

Different measures of agreement can be derived by counting the pairs of points in which the labellings agree or disagree. Specifically, any pair of data points falls

Table 2: The contingency table for correct labels U and cluster labels V.

U / V	v_1	v_2	$\dots$	v_C	Sums
u_1	n_{11}	n_{12}	$\dots$	n_{1C}	$n_{1\cdot}$
u_2	n_{21}	n_{22}	$\dots$	n_{2C}	$n_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_R	n_{R1}	n_{R2}	$\dots$	n_{RC}	$n_{R\cdot}$
Sums	$n_{\cdot 1}$	$n_{\cdot 2}$	$\dots$	$n_{\cdot C}$	n

into one of four categories: (I) the points have the same label in both U and V; (II) the points have different labels in both U and V; (III) the points have different labels in U but the same label in V; (IV) the points have the same label in U but different labels in V (Hubert and Arabie, 1985). For our interests, let N_I denote the number of pairs in category (I) and N_{II} the number of pairs in category (II). Now, by (Hubert and Arabie, 1985), these are given by

$$N_I = \frac{1}{2}\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}(n_{ij}-1), \tag{2.25}$$

$$N_{II} = \frac{1}{2}\left(n^2 + \sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2 - \left(\sum_{i=1}^{R} n_{i\cdot}^2 + \sum_{j=1}^{C} n_{\cdot j}^2\right)\right) \tag{2.26}$$

and the Rand Index can be presented as

$$\mathrm{RI} = \frac{N_I + N_{II}}{\binom{n}{2}}. \tag{2.27}$$

Effectively this is the probability of agreement of the two labellings and can be used as a clustering metric as itself. However, according to (Hubert and Arabie, 1985), this is not *corrected for chance*. This means that its expected value (value it gets under random labelling) is not necessarily near zero, or even constant, and the index value is thus hard to interpret. A good measure of similarity will take on some constant value under an appropriate null model of how the labels have been chosen.

Constructing such a null model starts by assuming that both U and V are picked at random given that both have the original number of labels and objects in each (i.e. they come from generalised hypergeometric distribution). Then we can calculate the Expected Rand Index as (Hubert and Arabie, 1985)

$$\mathbb{E}(\text{RI}) = 1 + \frac{2\sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}}{\binom{n}{2}^2} - \frac{\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}}{\binom{n}{2}}. \tag{2.28}$$

The general version of chance correction is

$$\frac{\text{Index - Expected Index}}{\text{Maximum Index - Expected Index}},$$

and so the corrected for chance Rand Index, or Adjusted Rand Index, has the form (assuming maximum to be 1 and after some manipulation) (Hubert and Arabie, 1985)

$$\text{ARI} = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}/\binom{n}{2}}{\frac{1}{2}\left(\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}\right) - \sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}/\binom{n}{2}} \tag{2.29}$$

This value is bounded above by 1 and equals to 0 when the index equals the expected values. Therefore, the value 1 corresponds to perfect labelling and values closer to 0 correspond to random labelling with respect to the given correct labels U.

2.2 Self-Organizing Map

As mentioned at the beginning of the chapter, SOM is a neural network used in dimensionality reduction highly capable in visualisation. SOM can be seen as both a projection and a clustering method: SOM projects data into a lower dimensional space in a nonlinear way by representing the input data by local averages (Kohonen, 2014, p. 11). SOM was developed by a Finnish academic Teuvo Kohonen in the early 1980s and since then has been used in a wide variety of fields; Kohonen himself gives an overview of SOM applications in (Kohonen, 2014, ch. 2). However, in mental stress detection context SOM has only been used by (Huysmans et al., 2018).

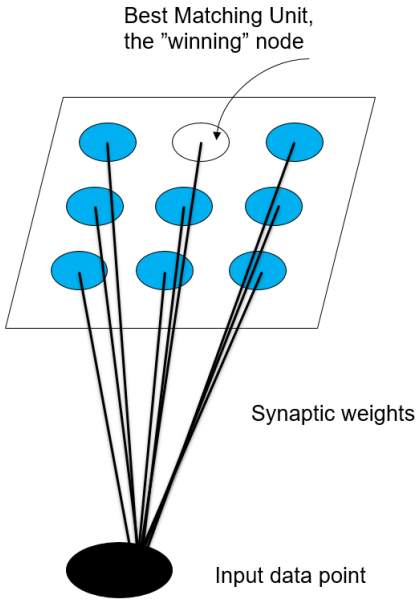


Figure 2: Structure of SOM. Each input data point is connected to each SOM neuron by a synaptic weight. One node is the "winner" , the most similar point in the map. The picture is modified from (Haykin, 2008, p. 427).

The goal of SOM is to transform the input data of arbitrary dimension into a low-dimensional (usually two-dimensional) map. To do this, SOM uses a grid-shaped single-layer feedforward artificial neural network, where each input data point is connected to each discrete point on the map, called *neurons*. The structure is depicted in Figure 2. The training process can be thought of as *competitive, cooperative* and *adaptive* (Haykin, 2008, p. 429-430). This means that when an input data point is presented to SOM, the neurons compete on which one is the most similar and the synaptic weights of the winning node, called *best matching unit* (BMU), and its closes neighbours are updated to match the data point even more.

In this section, we go through the mathematics behind the original SOM training algorithm and take a look at some clustering and visualisation options. Instead of the original paper (Kohonen, 1982), we follow the clearer and more thorough presentation given in (Haykin, 2008, ch. 9.3).

2.2.1 Training Algorithm

Let $\mathbf{x} \in \mathbb{R}^m$ be an input data point and

$$\mathbf{w}_l = [w_{l1}, w_{l2}, \ldots, w_{lm}]^T, \quad l = 1, 2, \ldots, L$$

the *synaptic weight vector* of neuron l, where L is the number of neurons and m is the number of dimensions. We define the most similar neuron as the one that minimizes the Euclidean distance between a synaptic weight and input, denoted by

$$i(\mathbf{x}) = \arg\min_l ||\mathbf{x} - \mathbf{w}_l||, \quad l \in \mathcal{A}, \tag{2.30}$$

where $||\cdot||$ denotes the Euclidean distance and $\mathcal{A}$ the lattice of neurons. Observe that now $i(\mathbf{x})$ is the BMU of sample $\mathbf{x}$, which concludes the competitive process.

At this stage of training, we have found the best-matching unit i for the input data point $\mathbf{x}$. Next, we must find a set of spatial neighbours of i. This gives rise to the need for a *neighbourhood function*, to determine the topological neighbourhood centered on winning neuron i, which is the set of neurons excited by the input data point.

Let $h_{l,i}$ denote a specific topological neighbourhood centered on winning neuron i and covering a typical excited neuron l, and let $d_{l,i}$ be the distance between l and i. A typical choice for the neighbourhood function is the Gaussian function (Haykin, 2008, p. 431),

$$h_{l,i(\mathbf{x})} = \exp\left(-\frac{d_{l,i}^2}{2\sigma^2}\right), \quad l \in \mathcal{A}. \tag{2.31}$$

Another option is to consider the c-neighbours of the neuron, where $c \in \{4, 6, 8\}$ depending on SOM structure. If the structure is rectangular like in Figure 2, the 4-neighbours of each point are all the immediate neighbours up and down, left and right and 8-neighbours also include the neurons to the upper and lower left and right. In a hexagonal grid, 6-neighbours of each neuron are all its immediate neighbours.

The distance $d_{l,i}$ in (2.31) can be any well-defined metric, but the usual choice is the Euclidean distance. In (2.31), σ is called *the effective width* of the neighbour-

hood and can be defined to be constant or some temporally decreasing function, like the exponential decay

$$\sigma(t) = \sigma_0 \exp\Big(-\frac{t}{\tau_1}\Big), \quad t = 0, 1, 2, \ldots \tag{2.32}$$

or linear decay

$$\sigma(t) = \frac{\sigma_0}{t}, \quad t = 0, 1, 2, \ldots. \tag{2.33}$$

Above, t is the number of iterations over the whole training data (epochs), σ_0 is the effective width at the beginning of training and τ_1 is a time constant chosen by the designer. If a decreasing function is used, the set of neurons updated at the beginning of training is naturally larger than towards the end of training (Haykin, 2008, p. 432). This concludes the cooperative process.

Now we have found the BMU $i(\mathbf{x})$ of an input data point $\mathbf{x}$ and a neighbourhood $h_{l,i(\mathbf{x})}$ of excited neurons l around $i(\mathbf{x})$. What is left is updating the weights $\mathbf{w}_l$ of all neurons in $h_{l,i(\mathbf{x})}$ in relation to the input vector $\mathbf{x}$ - this is actually what makes the process self-organizing (Haykin, 2008, p. 433).

At the beginning of training the weights are initialized according to some given procedure. Now, given the synaptic weight vector $\mathbf{w}_l(t)$ of neuron l at iteration $t = 1, 2, 3, \ldots$, the updated weight vector at iteration $t + 1$ is given by

$$\mathbf{w}_l(t+1) = \mathbf{w}_l(t) + \eta(t) \cdot h_{l,i(\mathbf{x})}(t)\big(\mathbf{x}(t) - \mathbf{w}_l(t)\big), \tag{2.34}$$

where η is the learning rate parameter defined below. This is applied to all neurons in the topological neighbourhood of winning neuron i. The derivation of this rule is presented in both (Kohonen, 1982; Haykin, 2008). This concludes the adaptive process and one iteration of the algorithm is done. Iterating over all the training samples once is called an *epoch*.

Equation (2.34) has the effect of moving the synaptic weight vector $\mathbf{w}_l$ towards the input data point $\mathbf{x}$, thus self-organizing the neurons in a way that similar neurons lie close to each other on the map. The learning rate parameter η can be set constant but like σ in (2.31), it is more customary to make it decrease in time according to some decreasing function, like the exponential decay

$$\eta(t) = \eta_0 \exp\Big(-\frac{t}{\tau_2}\Big), \quad t = 0, 1, 2, \ldots, \tag{2.35}$$

where τ_2 is another time constant.

This concludes one iteration of the SOM algorithm. Next, another input data point $\mathbf{x}'$ is introduced to the SOM, and the process is continued until convergence. This on-line version of the algorithm is summarised in Algorithm 2.

According to (Kohonen, 2014, p. 37), instead of the on-line sample-by-sample training of SOM, a batch version should be preferred. As explained in (Kohonen, 2001, p. 139-140), in the batch version we update the synaptic weights after a certain number of training samples have been presented to the map. The synaptic weight w_l is updated to be the average of all the points in this batch that are in its topological neighbourhood. Now, (2.34) can be expressed as

$$\mathbf{w}_l(t) = \frac{\sum_{t'=1}^{t} h_{l,i(\mathbf{x})}(t')\mathbf{x}(t')}{\sum_{t'=1}^{t} h_{l,i(\mathbf{x})}(t')}. \tag{2.36}$$

The formula (2.36) is from (Wittek et al., 2017), whose implementation of SOM training is used later. Note that the formula does not make reference to previous value of the weight, because the summation runs over epochs of training process.

To assess convergence of the map, (Kohonen, 2001, p. 313) says that the *quantization error* is a sensitive measure of the mapping accuracy. The quantization error is given by

$$Q(\mathbf{X}) = \frac{1}{n}\sum_{j=1}^{n} ||\mathbf{x}_j - i(\mathbf{x}_j)||, \tag{2.37}$$

where $\mathbf{x}_j$ is a single data point and $i(\mathbf{x}_j)$ its BMU. Thus the quantization error is the average distance between data points and their BMUs.

Algorithm 2 Self-Organizing Map

1: Initialize the synaptic weights $\mathbf{w}$.
2: Randomly sample an input data point $\mathbf{x}$ from the input data $\mathbf{X}$.
3: Find the best matching unit for the given data point using (2.30).
4: Update the synaptic weights of all the excited neurons (2.31) by the formula (2.34).
5: Repeat steps $2-4$ until convergence or a predefined number of epochs.

2.2.2 Visualisation & Clustering

The synaptic weights $\mathbf{w}_l$ of SOM are stored to a $L \times m$ matrix W called *codebook*; here L is the number of neurons and m is the number of dimensions in input data. If the codebook is organized as $r \times k \times m$ tensor, where r and k are the number of rows and columns in the map, each dimension $m_1, m_2, \ldots, m_m$ can be visualised with a heatmap depicting each neuron's weight on dimension m_i. These are called *component planes* and they show which features have the most significant influence on clustering and how features are correlated (Stefanovič and Kurasova, 2011).

The U-matrix method (Ultsch and Siemon, 1990) allows to visualise the whole codebook W in a single plot, depicting all the dimensions, or features, of $\mathbf{X}$ and allowing to perceive possible clusters in the data. Using notation from Section 2.2.1, the U-matrix for a size $r \times k$ map will have size $(2r-1) \times (2k-1)$ and

$$U(i) = \sum_{h_{l,i}} ||\mathbf{w}_i - \mathbf{w}_l||, \tag{2.38}$$

where i is a neuron and l is in the topological neighbourhood $h_{l,i}$. The rest of the matrix elements are obtained by interpolation. Thus the U-matrix is an inter-neuron distance image where high values stand for dissimilarities between neurons, denoting cluster borders (Ultsch and Siemon, 1990).

Since SOM organizes the neurons in a way that most similar data points map to the same neuron, it is natural to utilise the resulting mapping in clustering. The easiest way to do this is to consider each BMU as a separate cluster. However, this is hardly optimal because even with a relatively small 10×10 SOM we would end up with at most hundred clusters (not all neurons are necessarily BMUs) which is probably not realistic for the majority of cases. As hinted, the U-matrix offers a way to approximate the number of clusters, by assessing the number of areas limited by borders. If we think of a normal map, borders can be thought of as mountains while valleys between the mountains are clusters of similar data points. See the effect in Figure 3.

A more conclusive clustering of the original data can be obtained by fitting some clustering algorithm using the SOM codebook as its input data, as proposed by (Vesanto and Alhoniemi, 2000). During the training process the similarity of

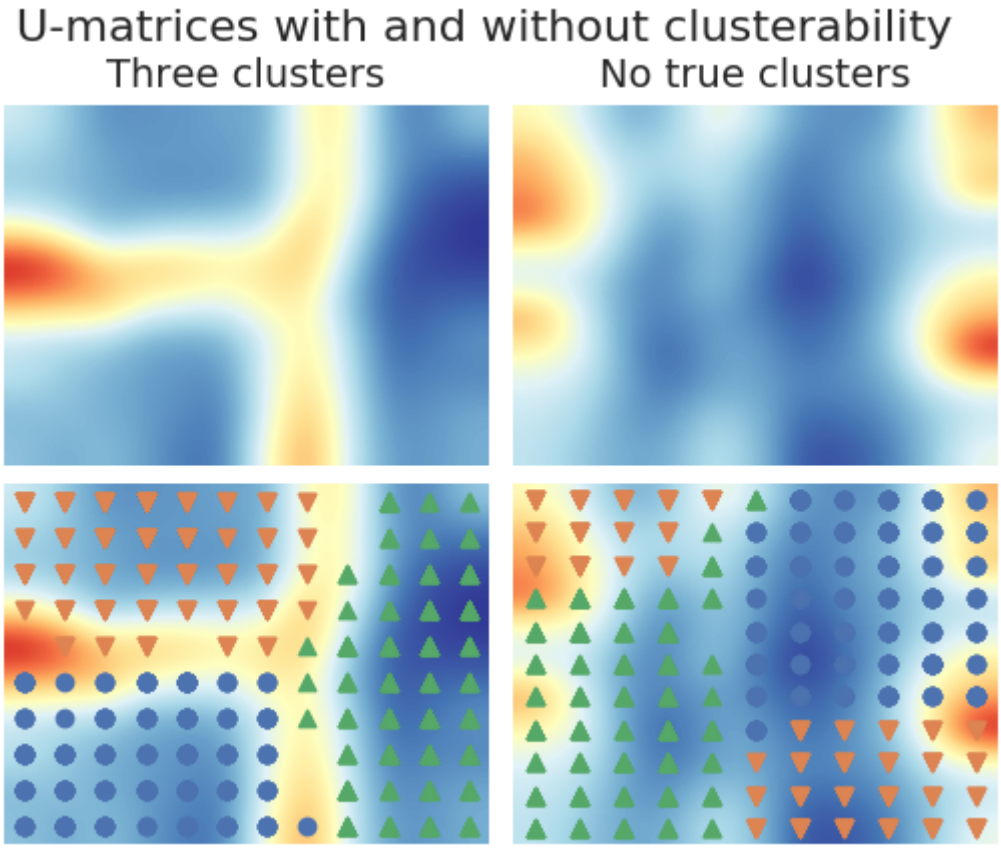


Figure 3: The U-matrices of SOMs calculated for similar datasets as in Figure 1. The yellow/red areas depict mountains and blue areas the valley regions. The clusters on the left are clearly distinguished and easily found by K-Means with K=3 on the lower left panel. On the right for the case of no clusters, K-Means still finds the clusters even though based on the U-matrix clusterability of the data is highly doubtful.

neighbouring codebook vectors (SOM prototypes, or the synaptic weights) is maximized and so the most similar neurons are already located close to each other. The resulting clustering can be plotted on the U-matrix which allows to visually estimate the overall data clusterability and whether the clustering was successful.

The resulting clustering can be used to predict the cluster of a new, unseen

data point. We first determine the BMU of the new point by Equation (2.30) and then check which cluster its BMU belongs to. The same cluster is assigned to the new observation.

2.2.3 Setting Hyperparameters

The SOM algorithm has a lot of moving parts and hyperparameters, like the map size and shape, map topology, neighbourhood and parameter decay functions. There is no true ground truth as to which parameters work best for a given data but we will shortly cover some of the options in the original SOM Toolbox for MATLAB (Vesanto et al., 2000). Their recommendations to various parameters are as follows:

- The number of neurons on the map should be around $M = 5\sqrt{n}$, where n is the number of observations.
- One side length of the map should be longer than the other, with aspect ratio given by the ratio of the eigenvalues of the training data.
- The shape of the map should usually be planar and the lattice hexagonal.
- They do not give firm recommendations for the neighbourhood function or the parameter decay function. They only state that the learning rate function and the effective width of the neighbourhood should decrease in time.
- Initializing the map can be done with random numbers or along the linear subspace spanned by two first principal components.
- The number of training epochs should be at least ten times the amount of samples in training data.

2.3 Models and Personalisation

There are a number of different models that can be built from the pieces discussed in this chapter. The skeleton of the model used in this study is depicted in Figure 4.

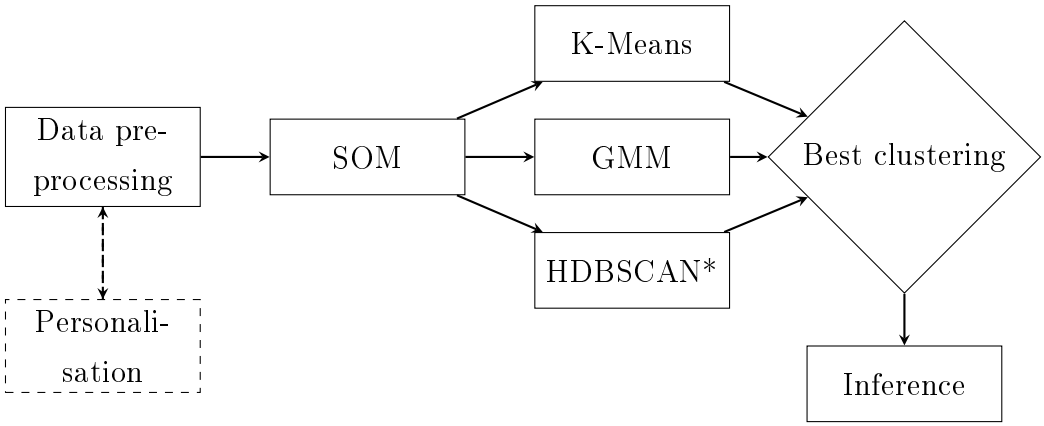


Figure 4: Pipeline for the stress detection algorithm. The dashed box "Personalisation" depicts an optional step.

As always, data preprocessing is a highly context-dependent step containing e.g. resampling, feature transformation, extraction and selection and accounting for missing data. As a next obligatory step, we calculate SOM. The map size and shape is set as recommended by (Vesanto et al., 2000) separately for each dataset fed to the system. There were no major differences in quantization error with the other SOM hyperparameters after experimenting with the real-life data used later on, and so they are kept at their default as provided by (Wittek et al., 2017): a planar rectangular map with Gaussian neighbourhood and linear learning rate and effective width decay. The SOM prototypes are clustered with three different algorithms the performance of which is estimated by the silhouette score (2.22). Unlike (Huysmans et al., 2018), we do not set a fixed number of clusters but look for $2 - 10$ clusters with K-Means and GMM, reducing the amount of supervision. The minimum cluster size for HDBSCAN* is context-dependent. Before inference, the best clustering is chosen according to the silhouette score and also the corre-

sponding U-matrices are drawn at this stage.

What to infer from the clusters found is again context-dependent. If we have ground truth labels, the cluster labels can be related to those via the confusion matrix by taking the row- or column-wise maximums and the performance can be assessed in terms of e.g. accuracy or the F1-score. The situation is somewhat more complex with real-life data when there are no sample-by-sample annotations available. Solutions to this are discussed in the analysis part of the study.

We have personalisation as an optional step in the pipeline. Usually, in any data-analysis process, we aim for as general model as possible, a model that is able to predict the outcome values of a new observational unit whose data it has not seen before. These models have no person-specific elements but as discussed in the Introduction, stress is a subjective feeling and therefore personalisation may produce better results in this context.

Broadly, three levels of personalisation can be distinguished: fully personal, semi-personal and general. Fully personalised models are fit separately for each individual. The semi-personal methods include e.g. adding a person-specific component to the model or fitting the model separately for groups of similar users. The general models are the most common ones, containing no person-specific elements.

Full personalisation has previously been employed by (Smets et al., 2016; Vildjiounaite et al., 2017, 2018). In this version, the model is fit using data of one subject only, and the performance is estimated by cross-validation or a single held-out test data. The training data consist of a subset of data from one individual and the test set is the rest of the data from the same individual.

There are several types of semi-personalised models used before in stress detection. (Shi et al., 2010) used a modified SVM with a person-specific component to build a model which benefited from the data of all individuals. A little similarly, (Taylor et al., 2017) made use of different Multitask Learning techniques that use all the data available but are customized to the needs of each individual. (Vildjiounaite et al., 2018) built their HMM model using data from all subjects but created a day-specific reference model used in actual stress recognition in a personalised way. In (Vildjiounaite et al., 2017; Xu et al., 2015), the authors

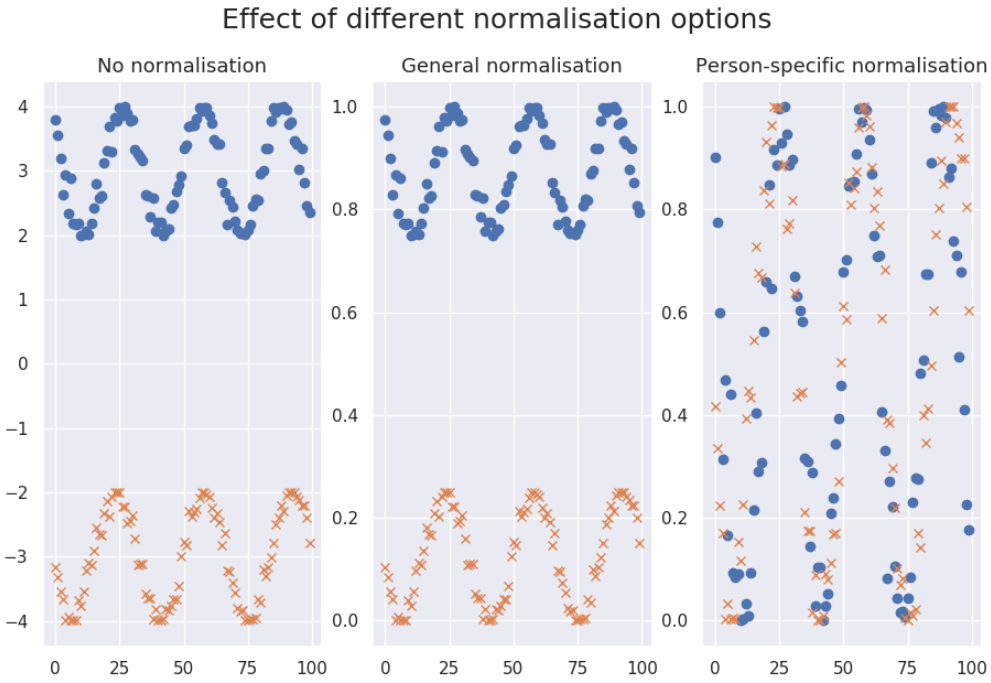


Figure 5: The effect of independent normalisation of time series. The left panel shows the original time series from two persons. In the middle a min-max normalisation (Eq. 3.1) with data from both persons is used. In the right panel, person-specific min-max normalisation is used, making the two time series look more identical.

first clustered the persons with K-Means clustering and afterwards trained cluster-wise models with leave-one-subject-out cross-validation, although, the former used within cluster cross-validation while in the latter one clustering was also part of the cross-validation scheme.

For the semi-personalised models in this work, we first cluster the people with K-Means, determining the correct number of clusters with the silhouette score. Then, we use a within-cluster leave-one-subject-out cross-validation to determine the model performance. In addition, we try another version that is related to feature normalisation. When building semi-personal or general models, the features can be normalised according to statistics calculated for all the data or each indi-

vidual's data separately. The latter may prove to be necessary because individuals' responses to different stimuli may differ and the model may find the underlying pattern easier. The two versions of normalisation are demonstrated in Figure 5. The within-subject normalisation has previously been employed by (Huysmans et al., 2018).

Lastly, we will consider non-personalised general models. However, general models will be fit separately both with general and person-specific normalisation, allowing for another level of personalisation. To summarise, the different levels of personalisation for SOMs are as follows:

Personal

- Fully personal model. SOM is fit using each individuals data only with personal normalisation.

Semi-personal

- Person-similarity model. Use K-Means to cluster persons and then fit cluster-wise SOM. Both with personal and cluster-wise normalisation.

General

- General model. SOM is fit with data from all the individuals with personal and general normalisation.

In this Chapter, we have presented the method for unsupervised stress detection. For each dataset, we calculate SOM, find the best clustering for it and relate the found clusters to stress in a data-specific way. Personalisation is done at two levels regarding how participants' data are combined for training and how the data are normalised. Next, we apply the method for two datasets, one from a laboratory setting and one with real-life data. Further details of the process and the exact models fit for each dataset are given in corresponding sections of Chapter 3.

3 Experiments

The laboratory data are an open-source dataset[1] for Wearable Stress and Affect Detection (WESAD) (Schmidt et al., 2018). It contains physiological and motion data from fifteen persons during a laboratory experiment in which the participants were subjected to different stimuli. The data come with both ground truth labels from the protocol and from self-reports. All the information told later regarding the dataset are from (Schmidt et al., 2018) or the readme-file attached to the data.

The real-life data was collected in a joint project of VTT, The Technical Research Centre of Finland, and FIOH, Finnish Institute of Occupational Health. The participants' daily activities were followed for four weeks with a Polar M600 smartwatch measuring heart rate, interbeat interval, and acceleration. In addition, their smartphone behaviour like application usage and screen on times were recorded and they got pop-up questionnaires three times a day, used as ground truth in the analysis.

All the analyses were done with the Python programming language. SOMs were calculated with the package somoclu (Wittek et al., 2017) and the clustering was done with the implementations in scikit-learn (Pedregosa et al., 2011) (K-Means and GMM) and hdbscan (McInnes et al., 2017) (HDBSCAN*). All the code for the analysis of the WESAD data was written by the author. For the real-life data, most of the code needed for the analysis was written by the author but some data preprocessing steps were written by the team at VTT.

3.1 Laboratory Data

Our analysis of the WESAD data follows the pipeline given in Figure 4. First, we give a short description of the data at hand and in subsequent sections, we walk through the pipeline to compare stress prediction capabilities of different models.

The WESAD data contain measurements with two devices, a chest-worn RespiBAN, and a wrist-worn Empatica E4. The modalities collected with both devices are pre-

[1]The data are available for download on **University of Siegen website**.

Table 3: The modalities available in the WESAD data.

Device	Sensor	Sample Rate (Hz)
RespiBan (chest)	Electrocardiogram (ECG)	700
	Electrodermal activity (EDA)	700
	Electromyogram (EMG)	700
	Respiration	700
	Body temperature	700
	Three-axis acceleration	700
Empatica E4 (wrist)	Blood volume pulse (BVP)	64
	Electrodermal activity (EDA)	4
	Body temperature	4
	Three-axis acceleration	32

sented in Table 3. For descriptions of different signals, we refer to (Sharma and Gedeon, 2012).

The authors recruited a total of 17 participants but data of two of them had to be discarded due to sensor malfunction. The remaining ones had a mean age of 27.5 years with standard deviation of 2.4 years and there were twelve male and three female subjects. The goal in their study was to elicit three different affective states, neutral, stress, and amusement. The laboratory protocol began with a twenty minutes long baseline measurement, during which the subjects were reading neutral material either standing or sitting. During the amusement condition, they were shown eleven funny videos, totalling around six and a half minutes. At the stress condition, the participants had to give a public speech and solve mental arithmetic tasks. The stress phase had a total length of around ten minutes. After these, the subjects went through a guided meditation session. The order of amusement and stress conditions were interchanged between subjects. The evolution of different sensor values over the study protocol is presented in Figure 6.

The authors provided a benchmark for the stress prediction task using 16 differ-

ent feature combinations derived from original sensor data and five different classifiers, all of which were supervised algorithms. In a three-class (baseline, stress, and amusement) problem, the best accuracy obtained was 80% with the AdaBoost classifier, using the physiological features of the chest-worn device. The corresponding F1-score was 72.5%. Overall, the highest accuracy scores ranged around $75\% - 80\%$ and were usually reached with combinations of all the available modalities, with wrist and chest features separately or together. Further information on used features and scores obtained can be found in tables 1 and 3 of (Schmidt et al., 2018).

3.1.1 Data Preprocessing

As provided by the authors, we use the part of the data that contains both devices' data with synchronised timestamps. Like them in their benchmark, we consider the data observed during each of the conditions (baseline, stress, amusement) and disregard the data with other labels. The left out data consist of transition and meditation periods, and labels which the authors say to ignore. Like the authors, we use the study protocol labels as ground truth.

The authors calculated a total of 80 features from the original signals in sliding windows with a window shift of 0.25 s. The actual length of the windows differed for different features. In our analysis, we skipped this computationally heavy step and used the raw sensor values resampled to 1 Hz by mean aggregation. This was done to compress the data to a smaller number of observations and thus to lower the computational cost, and to demonstrate the pattern recognition capability of our model by working only with as simple and small number of features as possible. However, we used data from both wrist and chest devices but only separately.

This way, we had a dataset of approximately 2200 observations with 6 and 8 columns for wrist and chest data, respectively, for each participant. The data amounted to approximately 37 minutes and contained on average 53% of baseline data, 30% of stress data and 17% of amusement data. Therefore, we had an imbalanced label distribution.

We used the min-max normalisation (Pedregosa et al., 2011), which transforms

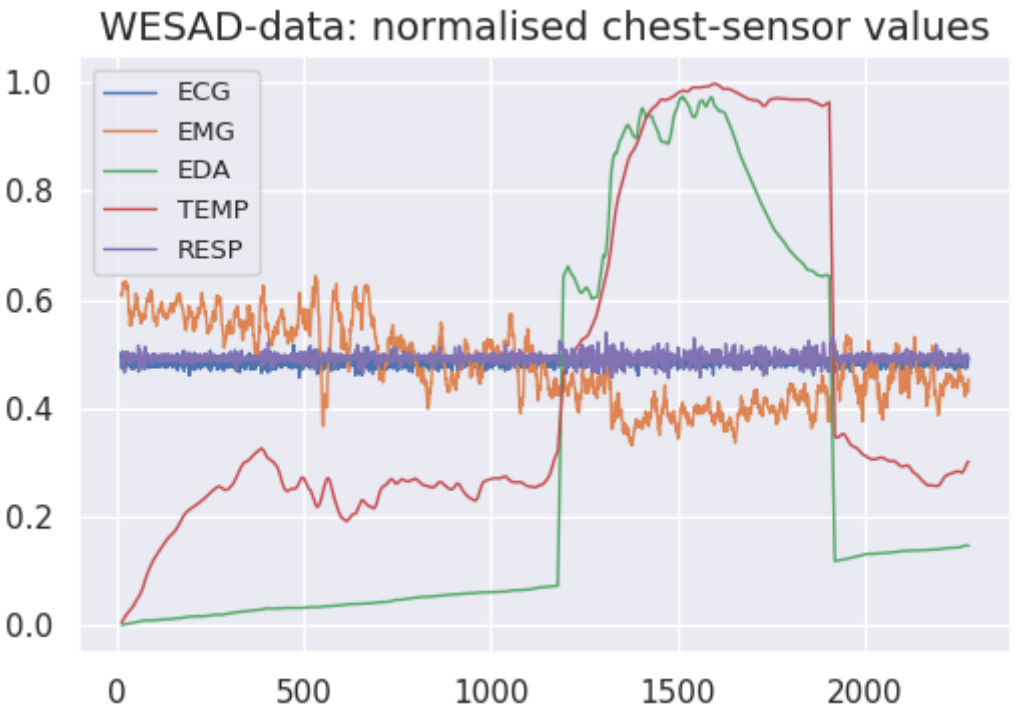


Figure 6: The minmax-normalised values from the physiological sensors of the chest-device for one user. The condition changes can be seen around timestamps 1200 and 1900. The lines depict running 15-second means. As is apparent, the sensor values vary between conditions, most notably the electrodermal activity and body temperature (green and red lines). The middle condition also shows more variation in respiration (purple) and lower values in electromyogram (orange).

the variables to range $[0, 1]$, and is given by

$$\frac{\mathbf{x}_p - \min \mathbf{x}_p}{\max \mathbf{x}_p - \min \mathbf{x}_p} \tag{3.1}$$

for feature $\mathbf{x}_p$. The minimums and maximums were taken to be person-specific, cluster-specific or general, depending on the level of personalisation. The different personalisation schemes were discussed in Section 2.3.

As part of person-similarity models, we clustered the persons with K-Means clustering. To do this, we used the data from the chest-worn device and calculated user-wise means and standard deviations (stds) for each modality. For the three

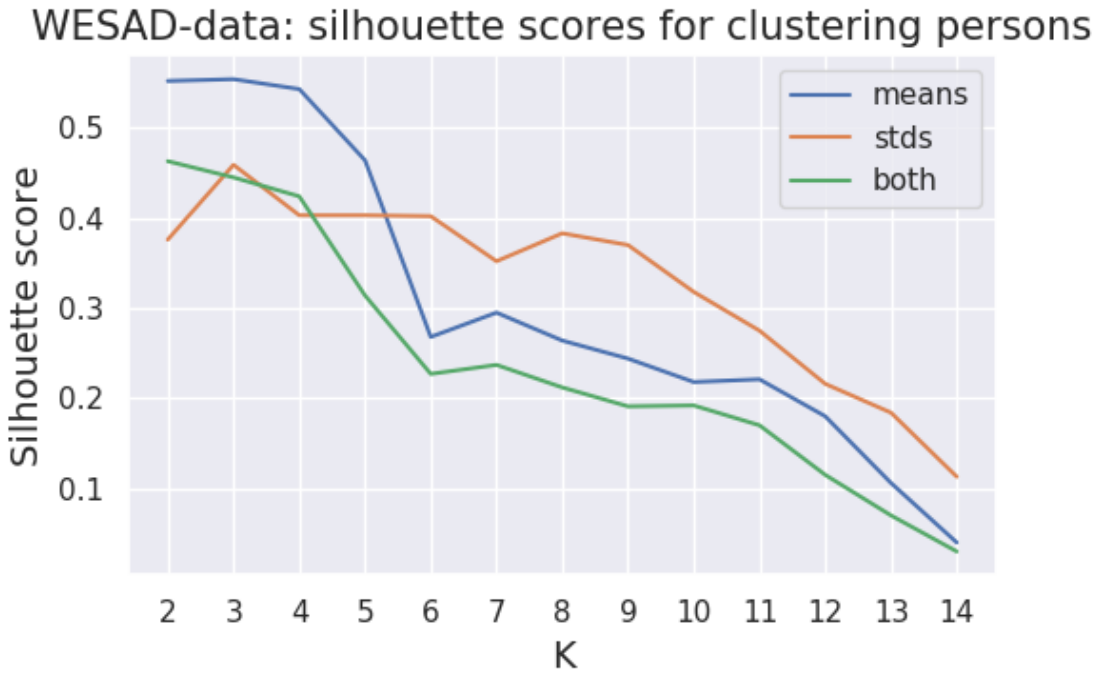


Figure 7: The silhouette scores along different number of clusters and different sets of features used in person-similarity clustering of WESAD data.

combinations of these features (means, stds, means + stds), we fit K-Means with K ranging from 2 to 14. The silhouette scores observed are depicted for each feature combination in Figure 7. The highest silhouettes are scored at $K = 2$ and $K = 3$. Starting from $K = 3$ at least one of the clusters would contain only one subject and to prevent this, we chose $K = 2$. The division with ID's provided by the authors was as follows:

cl1: S3, S5, S6, S7, S11, S17

cl2: S2, S4, S8, S9, S10, S13, S14, S15, S16

For validation of the results, we used leave-one-subject-out (LOSO) cross-validation for the semi-personal and general models. This means that we trained the model with all the other users' data and left out one user's data for testing. The evaluation score now describes how well the model generalises and performs

for a new subject whose data it has not seen before. Because the personal model used data from just one individual, this procedure could not be used. Instead, we left out 20% from the end of each condition. The prediction score now describes how well the model generalises for new data from that individual. In both cases, the correct clusters for the test data were obtained by projecting the data onto the SOM by Equation (2.30) and assigning each test set data point to the cluster its BMU belonged to. The required data splitting for these two validation procedures was done at the preprocessing step, before starting the SOM calculations.

3.1.2 SOM and Clustering

There were a total of 150 different data matrices coming out of the data preprocessing step. The number constituted of

1. Data of each individual for personalised models, for both wrist and chest -device. $(15 \times 2 = 30)$

2. Data from both user-clusters with each subject's data left out in turn, for both wrist and chest -device, both with personal and cluster-wise normalisation. $((6 \times 2 + 9 \times 2) \times 2 = 60)$

3. Whole data from all the subjects with each subject's data left out in turn, for both wrist and chest -device, both with personal and general normalisation. $((15 \times 2) \times 2 = 60)$.

The SOM topology was selected separately for each dataset as per discussed in Section 2.2.3. The number of neurons was set to around $5\sqrt{2200} \approx 235$ for individual SOMs, around $5\sqrt{5 \cdot 2200} \approx 525$ and $5\sqrt{8 \cdot 2200} \approx 665$ for user-clusters $cl1$ and $cl2$ and to around $5\sqrt{14 \cdot 2200} \approx 880$ for general SOMs. The aspect ratio was set to the square root of the ratio of two largest eigenvalues of each data matrix. The SOMs are trained for 1000 epochs and all the other SOM parameters are set to the defaults of the package somoclu (Wittek et al., 2017).

After training the SOM, its prototypes were clustered in turn with K-Means, GMM, and HDBSCAN*. The number of clusters for K-Means and GMM was

ranged from $2 - 10$ and the minimum cluster size for HDBSCAN* was ranged from $0.05 \cdot L$ to $0.9 \cdot L$ in steps of 10, where L is the number of neurons. With GMM, each point was assigned to the cluster with the highest probability. The clustering was evaluated by the silhouette score and the best parameters for each method were saved. In the end, the SOM codebook was clustered with the best method. However, because now we know that all the observations should be in a cluster, we required HDBSCAN* to cluster at least 80% of the data or it would not be chosen the best option.

The analysis cycle was repeated ten times to account for the stochastic nature of the model. The clustering results over all the runs are presented in Table 4. The statistics included are average and standard deviation of number of clusters found (Clusters), silhouette score (Silhouette), proportion of data clustered (Clustered (%)), and the total ratio of times found to be the best method (Best (%)) and the total ratio of times HDBSCAN* was able to cluster at least 80% of the data (Av (%)). These are reported across all different combinations of personalisation, the device the data come from and the method used for clustering the SOM prototypes. In addition, the statistics are reported separately for HDBSCAN* when it clustered enough data. This never occurred for some personalisation-device combination and so the corresponding rows are left out of the table.

After finding the best clustering for each SOM, we related the cluster labels to the correct labels via the confusion matrix, choosing the mapping so as to maximise the resulting *training* accuracy. This was the only part of the process we used the correct labels for and it adds some supervision to the procedure but a fully unsupervised version can be obtained by fixing the mapping to row- or column-wise maximums of the confusion matrix between the cluster and the correct labels. Because this method of relating cluster and true labels to each other does not penalise the number of clusters found, we also use ARI for estimating the prediction capability. The prediction performance over the ten runs is presented in terms of accuracy, F1-score and ARI in Table 5.

Table 4: Clustering results for the WESAD data. The values are means of the number of clusters, the silhouette score, and the amount of data clustered with standard deviation in parentheses, the proportion of times chosen as best method, and the proportion of SOMs the method was available (Av). Empty cells in the column Av denote "always".

Model	Device	Method	Clusters	Silhouette	Clustered (%)	Best (%)	Av (%)
PERS	chest	GMM	2.8 (1.0)	0.56 (0.12)	100	8	
		K-M	2.9 (0.9)	0.57 (0.12)	100	57	
		HDBS	2.2 (0.6)	0.66 (0.10)	72 (17)	0	
		$HDBS_{\geq 0.8}$	2.1 (0.5)	0.72 (0.10)	88 (5)	35	37
	wrist	GMM	4.6 (2.5)	0.57 (0.07)	100	7	
		K-M	5.0 (2.5)	0.59 (0.07)	100	81	
		HDBS	2.3 (0.8)	0.69 (0.09)	67 (14)	0	
		$HDBS_{\geq 0.8}$	2.5 (0.9)	0.58 (0.12)	91 (7)	11	13
PSPN	chest	GMM	3.9 (2.5)	0.32 (0.08)	100	2	
		K-M	3.7 (1.8)	0.37 (0.05)	100	98	
		HDBS	1.7 (1.1)	0.38 (0.24)	30 (24)	0	
	wrist	GMM	8.1 (1.9)	0.31 (0.03)	100	0	
		K-M	9.1 (0.9)	0.35 (0.02)	100	100	
		HDBS	2.0 (0.7)	0.48 (0.15)	34 (13)	0	
PSCN	chest	GMM	4.2 (2.7)	0.38 (0.08)	100	4	
		K-M	2.7 (1.1)	0.42 (0.05)	100	79	
		HDBS	1.8 (0.8)	0.47 (0.20)	50 (29)	0	
		$HDBS_{\geq 0.8}$	2.0 (0.0)	0.51 (0.09)	87 (5)	17	18
	wrist	GMM	7.0 (2.6)	0.39 (0.02)	100	2	
		K-M	8.4 (1.6)	0.42 (0.01)	100	98	
		HDBS	2.2 (0.5)	0.52 (0.07)	48 (13)	0	
GPN	chest	GMM	2.9 (0.7)	0.26 (0.02)	100	0	
		K-M	3.8 (0.5)	0.30 (0.01)	100	100	
		HDBS	0.8 (1.0)	0.22 (0.27)	7 (9)	0	
	wrist	GMM	7.4 (2.0)	0.24 (0.01)	100	0	
		K-M	9.1 (0.9)	0.29 (0.01)	100	100	
		HDBS	1.7 (0.8)	0.38 (0.18)	32 (15)	0	
GGN	chest	GMM	3.7 (2.2)	0.33 (0.02)	100	11	
		K-M	3.4 (1.6)	0.35 (0.01)	100	89	
		HDBS	0.5 (0.8)	0.10 (0.18)	10 (21)	0	
	wrist	GMM	6.2 (3.0)	0.33 (0.01)	100	0	
		K-M	7.3 (2.2)	0.36 (0.01)	100	100	
		HDBS	2.0 (0.1)	0.52 (0.04)	51 (8)	0	

PERS: personal, PSPN/PSCN: person-similarity with personal/cluster-wise normalisation, GPN/GGN: general with personal/general normalisation, K-M: KMeans, HDBS: HDBSCAN*, $HDBS_{\geq 0.8}$: HDBSCAN* when it clustered at least 80% of the data.

3.1.3 Results and Discussion

Comparing the clustering performance of different models, it seems that the fully personal model was able to find clusterings with the highest silhouette scores. Regardless of the measurement device and clustering method used, they have the silhouette score in the range $0.56 - 0.72$, reflecting dense and separated clusters. All the other personalisation schemes score silhouettes mostly around $0.30 - 0.40$, with the lowest score as low as 0.1 and even the best at 0.52, which is lower than the lowest score with fully personalised models. However, the standard deviations are generally higher with fully personalised models than with the other methods, hinting a large variation in the quality of clustering between different individuals. Regardless of the method, nearly all of the silhouettes are well above zero, meaning most samples were assigned to the correct cluster, and well comparable to that obtained by (Huysmans et al., 2018), whose silhouette score with a general model and personal normalisation using LOSO-validation was 0.301 with standard deviation of 0.0152.

The distribution of silhouette scores across subjects and training iterations is shown in Figure 8. We see that the within-subject deviation between training iterations is generally small, telling that similar patterns are found regardless of the iteration. The between-subject deviation is higher and medians range from around 0.43 to 0.80. The silhouettes found based on different device data are usually around the same range (within-subject) but there are large differences for some subjects (e.g. S11 and S14).

Overall the amount of clusters found by K-Means and GMM seems to be biased upwards, especially with the data from the wrist device where $7 - 9$ clusters are usually found. The situation is clearly better with the chest data with both of the clustering methods finding the expected $3-4$ clusters with all the models. This may suggest that the data from the wrist device are noisier and more inconsistent, and different anomalies caused by this are found instead of true behaviour patterns.

In contrast, the amount of clusters found by HDBSCAN* is biased downwards. It almost never finds more than two clusters and there is a downward trend in clusters found by it as the level of personalisation gets lower. However, HDBSCAN*

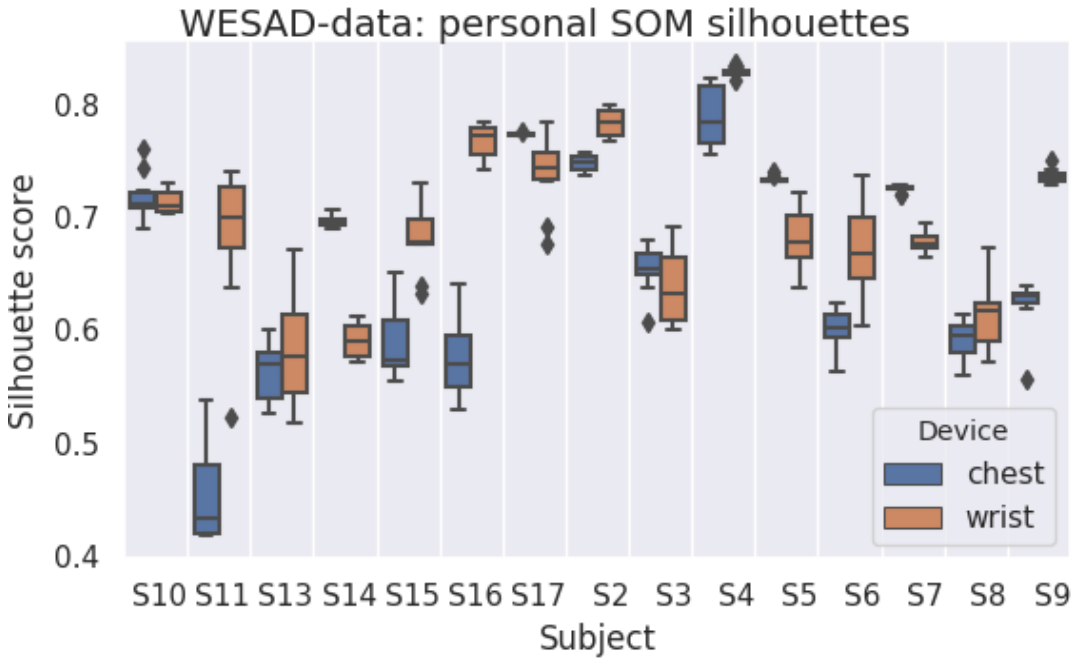


Figure 8: Subject-by-subject boxplots of the silhouette scores over the WESAD-data training iterations for the personal SOMs.

is able to cluster at least 80% of the data for only three model-device combinations, two of which are at a fully personal level. The usual amount of data clustered varies a lot between different models, generally being somewhere between $30-50\%$. The standard deviation of the amount of data clustered seems to always be quite high, suggesting big differences between different SOMs. Interestingly, HDBSCAN* is nearly always the best method when it clusters the required amount of data. Even when it doesn't, its silhouette scores are higher than with the other two methods. Albeit noteworthy, this is not surprising since it is easier to be more often right if you make fewer assignments.

Using person-specific normalisation produced slightly lower silhouettes than its conventional counterpart. This may be due to the LOSO-validation scheme, in which we attempt to use the trained model for a new person. Because the individual responses to stimuli differ, the same values with person-specific normalisation may mean different actual sensor values and thus the model may not find the underlying

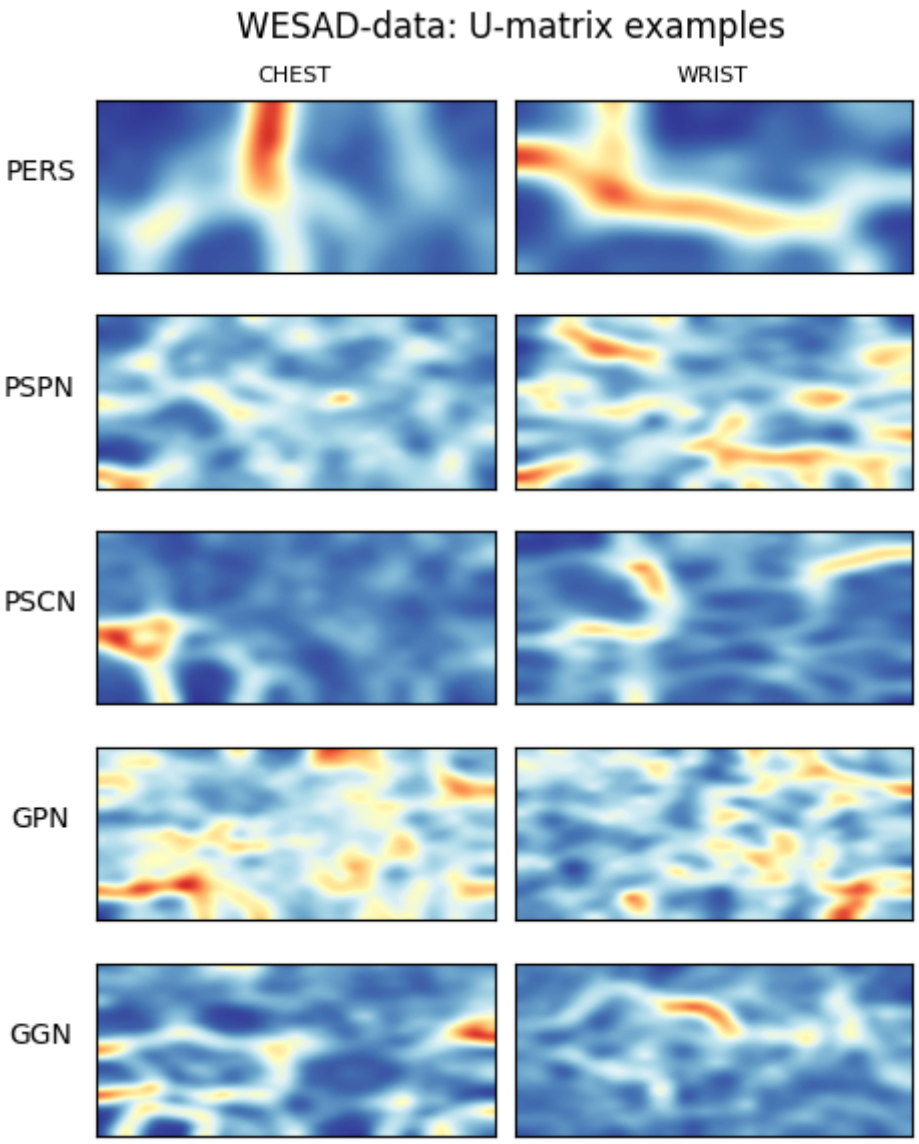


Figure 9: Examples of U-matrices drawn for WESAD data at each level of personalisation with both devices.

patterns for the new person. Had we done a train-test-split similar to the one with fully personalised model and left out 20% of each participants' data for testing, and trained the model with all the rest of each participants' data, the situation might be different.

Based on these results, the best clustering method to combine with SOM is K-Means. At all levels of personalisation, K-Means is most often chosen over the other two methods and it is always chosen for four out of ten model-device combinations. Therefore estimating the covariance structure with GMM doesn't seem to bring any added value. Since this is the first investigation of combining SOM with HDBSCAN*, it is difficult to say why the density based method mostly labelled data to noise. One would expect that clusters in SOM are dense because the map is organized so that most similar points are close to each other. It may be, however, that they are not dense enough or that there are no true clusters to be found.

The latter conclusion is given credibility in the U-matrices plotted in Figure 9. At the personal level, U-matrices originating from both chest and wrist device show clear borders between blue areas, with the chest data U-matrix showing three, maybe four clusters and the wrist data one showing four clusters. At all the other levels of personalisation, the U-matrices are inconclusive and do not show clear cluster boundaries. By the amount of "mountains" separating the "valleys", it would seem that there may be some underlying structure that is not grasped by the SOM. These remarks are in line with (Huysmans et al., 2018), who reported that no clear conclusions could be drawn from the U-matrix method.

The prediction results shown in Table 5 correspond quite well to the remarks done previously. Similarly to the silhouette score, the highest scores are obtained with personalised models, with accuracy and F1-score topping to 0.92 and 0.89, respectively. However, as pointed out previously, these two measures are now biased because they are calculated by mapping the cluster labels to the correct labels by maximising training accuracy and they are not penalised for an incorrect number of clusters found. They are only reported here because they are standard measures used previously e.g. by the WESAD dataset authors (Schmidt et al., 2018), whose highest accuracy scored was 0.80 in the three class problem. A more robust value is the Adjusted Rand Index, but its value cannot be straightforwardly compared to those obtained in previous work.

Now ARI tops to 0.80 with the fully personalised model but tends to get relatively high standard deviations telling that there is much variation between differ-

ent SOMs. We also note that the person-similarity model with personal normalisation scores $0.49 - 0.53$ and the same model with cluster-wise normalisation scores ARI of $0.23 - 0.34$. The same phenomenon is seen with the general model: personal normalisation scores are between $0.47 - 0.61$ and general normalisation scores are $0.21 - 0.24$. As evidenced in Table 4, the silhouette scores for both person-similarity and general models were a little higher with the more general normalisation, which is opposite to what was found in the prediction task. While the differences are little in silhouettes, the ARI is clearly higher when using the person-specific normalisation. Therefore it looks like personalised normalisation does not improve the actual clustering but allows to generate more reliable predictions.

The subject-by-subject distribution of ARI-scores over the training iterations is shown in Figure 10. The prediction scores seem to be somewhat more individual

Table 5: The prediction results for the WESAD data when using the best clustering for each SOM based on silhouette score. The values are means with standard deviations in parenthesis.

Model	Device	Clusters	Accuracy	F1-score	ARI
PERS	chest	2.7 (0.9)	0.89 (0.10)	0.86 (0.13)	0.80 (0.21)
	wrist	3.8 (1.8)	0.92 (0.09)	0.89 (0.13)	0.76 (0.21)
PSPN	chest	3.7 (1.8)	0.73 (0.15)	0.66 (0.16)	0.53 (0.22)
	wrist	9.0 (0.9)	0.70 (0.16)	0.66 (0.15)	0.49 (0.16)
PSCN	chest	2.7 (1.1)	0.51 (0.14)	0.37 (0.14)	0.23 (0.27)
	wrist	7.2 (2.8)	0.54 (0.20)	0.46 (0.21)	0.34 (0.25)
GPN	chest	3.8 (0.5)	0.76 (0.09)	0.69 (0.09)	0.61 (0.25)
	wrist	9.0 (0.9)	0.70 (0.14)	0.62 (0.15)	0.47 (0.15)
GGN	chest	3.4 (1.6)	0.51 (0.08)	0.36 (0.09)	0.21 (0.25)
	wrist	6.4 (2.9)	0.56 (0.13)	0.45 (0.14)	0.24 (0.21)

PERS: personal, PSPN/PSCN: person-similarity with personal/cluster-wise normalisation, GPN/GGN: general with personal/general normalisation.

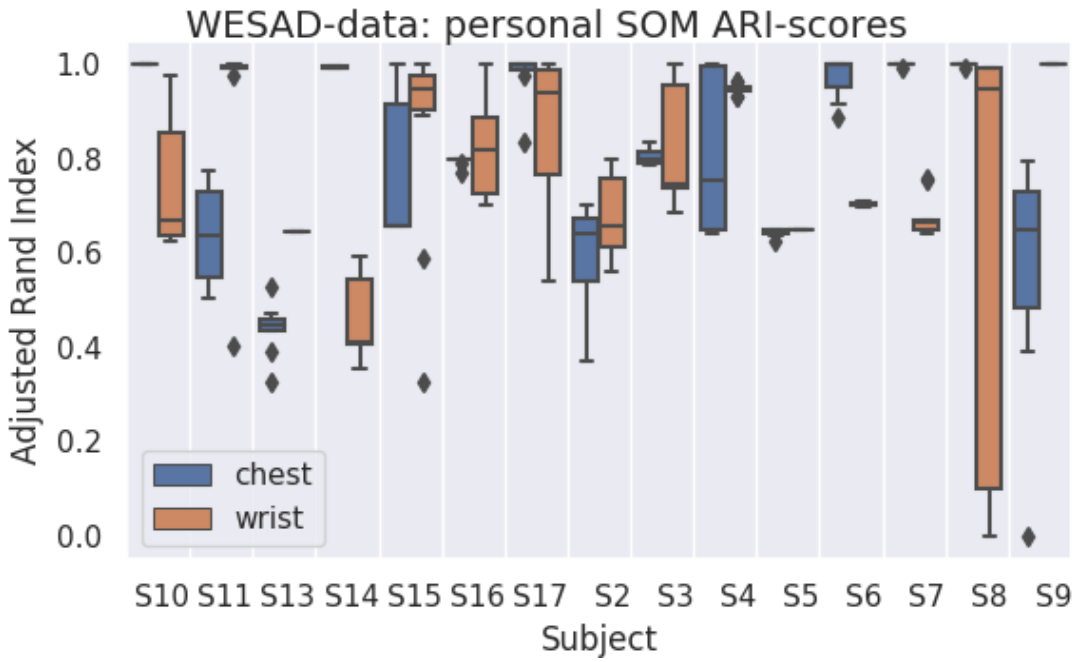


Figure 10: Subject-by-subject boxplots of ARI scores over the WESAD-data training iterations for the personal SOMs.

than the silhouette scores in Figure 8. The within-subject deviation for both devices and the difference between devices is small for some and high for others. The wrist-device scores for the subject labelled as "S8" seem to get values throughout the scale. After further investigation, the score was close to zero whenever the method found $2-4$ clusters and close to one when it found more. This suggests that when an almost correct number of clusters was found, they were highly confused as to which correct label they correspond to. However, overall the differences between training iterations seemed to be reasonable.

In their work, (Huysmans et al., 2018) used GMM with two components to detect between relaxed and stress phases and scored accuracy of 0.79. For further comparison with their work, we clustered all the SOMs generated over the ten runs of the analysis cycle with K-Means, setting $K = 3$ instead of using an inferred number of clusters. K-Means was chosen over GMM because in our analysis it has shown better performance than GMM. The prediction results for these runs are

Table 6: The prediction results for the WESAD data when using K-Means with fixed $K = 3$ for each SOM. Accuracy, F1-score and ARI are means with standard deviation in parentheses.

Model	Device	Clusters	Accuracy	F1-score	ARI
PERS	chest	3	0.89 (0.08)	0.86 (0.11)	0.81 (0.17)
	wrist	3	0.88 (0.14)	0.85 (0.15)	0.81 (0.21)
PSPN	chest	3	0.71 (0.15)	0.64 (0.16)	0.54 (0.22)
	wrist	3	0.68 (0.13)	0.59 (0.16)	0.37 (0.25)
PSCN	chest	3	0.55 (0.15)	0.42 (0.17)	0.34 (0.27)
	wrist	3	0.55 (0.10)	0.41 (0.12)	0.16 (0.25)
GPN	chest	3	0.77 (0.09)	0.69 (0.09)	0.61 (0.25)
	wrist	3	0.71 (0.12)	0.63 (0.14)	0.51 (0.19)
GGN	chest	3	0.51 (0.08)	0.35 (0.09)	0.23 (0.26)
	wrist	3	0.53 (0.02)	0.38 (0.03)	0.07 (0.14)

PERS: personal, PSPN/PSCN: person-similarity with personal/cluster-wise normalisation, GPN/GGN: general with personal/general normalisation.

shown in Table 6. The personal model seems to score higher and the general model with personal normalisation seems to score around the same as (Huysmans et al., 2018) did. In addition, all these scores are almost identical to those presented in Table 5, and so we may deduce that inferring the number of clusters automatically worked just as well as giving the correct number for the system.

In this section, we have demonstrated the use of SOM in stress/affection detection and compared different clustering and personalisation approaches. Overall the best scores were obtained with the fully personalised model, followed by person-similarity and general models with person-specific normalisation, and the best clustering method to combine with SOM was K-Means. The detection method described is unsupervised but manages to get stress/affection detection scores comparable to those obtained with supervised algorithms. The analysis conducted con-

solidates the capability of SOM in stress/affection detection and the results back up the previously presented impression that personalisation is needed in stress detection.

As the purpose of this analysis was solely to present the method and the ease of analysing laboratory data and we wanted to come by with as little features as possible, we did not consider feature extraction or selection at all. Employing this step to the analysis may increase the model performance. In addition, we trained each SOM for 1000 epochs, which is less than is thought to be ideal. As (Huysmans et al., 2018) trained their SOMs for 400 epochs, the effect of a longer training period should be investigated.

The main drawback in our approach is that any number of clusters may be found, and relating the found clusters to underlying ground truth is somewhat ambiguous. In the next section we see how the method is generalised to real-life data.

3.2 Real-Life Data

Similarly to the laboratory data, our analysis of the real-life data follows the pipeline given in Figure 4. The biggest differences are in data preprocessing and inference stages that now require more effort than was needed with WESAD data.

In this Section, we firstly give a brief description of the data and how it was collected. We discuss its quality and the problems related to gathering such real-world data and obtaining the ground truth. We present a method for converting the findings from SOM clustering to stress and then go through the analysis steps.

3.2.1 Data Description and Quality

The data contain four weeks of participants' daily activities collected with a Polar M600 smartwatch and their (working) smartphone. The data were collected between January and June 2018. The smartwatch data consist of heart rate, interbeat interval (IBI, the time between heartbeats) and three-axis acceleration. The smartphone data contain location, messaging and application usage, and the

status of both devices' battery and screen state were recorded. Before starting the data collection the participants answered a prequestionnaire concerning their demographics, health and working conditions. As compensation, they received the smartwatch used in data collection.

In addition to continuous measurements, the data contain subjects' answers to pop-up questionnaires appearing three times a day, at 9 am, 4 pm, and 9 pm. In the questionnaires they were asked to fill in their level of liveliness / sleepiness, calmness / nervousness, excitement / boredness, feeling of control and feeling of recovery in a 1 - 7 Likert-scale. They were also asked what they were mainly doing for the past 30 minutes, out of the 16 options given. Note that the level of stress was not asked directly.

An Android mobile application was developed for data gathering. The data collected with the smartwatch was transferred to the phone via Bluetooth and then all the data were continuously forwarded to a cloud-based server. On the server, the data were split up to different data types and converted to relational tables indexed by timestamps.

All the data were pseudonymised and no individual could be identified from the variables available. In addition, location and application data were anonymised by categorisation. The locations were transformed to contain only indicators of usual places instead of GPS coordinates. The application categories were formed after Google Play Store categories, and the categories used were business (e.g. calendar), communication (messaging apps), entertainment (games), infotainment (news), shopping (online shopping apps), social (social networks), travel / navigation (maps), utility (settings, updates), wellbeing (health), other and unknown. For the application usage and screen on times, we only stored the timestamp at which an app came to the foreground and went to the background and the time when the screen was turned on or off.

A total of 74 participants provided at least one day of data. Their mean age was 45.6 years with standard deviation of 9.9 years and there were 45 women and 28 men; one person did not answer the pre-questionnaire. All the participants were office workers and most of them (90%) were working full time. However, due to

problems with data quality much of the data had to be discarded.

The most important features we could obtain from the smartwatch were heart rate variability (HRV) measures derived from IBI, which have been found to be indicative of stress (Sharma and Gedeon, 2012). Correct and accurate IBIs can only be obtained from the ECG signal measured with a chest-belt but such continuous data collection would be tiresome and irritating. Wrist-worn devices offer a more unobtrusive alternative. However, the wrist devices calculate IBI from photoplethysmography-signal (PPG) which is sensitive to disturbance and improper attachment. As shown in (Pietilä et al., 2017), wrist-worn devices find the correct IBI well when the subject does not move but the performance gets worse during hand movements. The device we used was not tested in their study but the overall impression in our data was similar.

After applying a rolling mean filter to the IBI signal to detect false and correct values, hardly any of the daytime IBIs were correct and the IBIs found were artefacts caused by movement. During the night the situation was better but even then we could use data from 35 participants. This sums to 441 nights, with on average 12.6 nights (with standard deviation of 4.3 nights) per participant available. An often used HRV measure is the root mean square of successive IBI differences (RMSSD, see Eq. (3.4)), calculated over 5 minute periods (Shaffer and Ginsberg, 2017). The inclusion criterion was that for at least seven nights the participant provided at least two hours of RMSSD data. By night, we mean the time between 12 am and 6 am.

The phone data did not suffer from similar quality problems but we still had to discard some of the data. As explained shortly, we detected stress on a daily basis, and therefore we considered just the days with a 100% phone data coverage during daytime (between 6 am and 12 am). To have reliable ground truth we further required that at least two questionnaires were answered per day, and we only accepted subjects who provided at least seven days of data with these requirements.

This leaves us with 65 participants, totalling 1008 days with on average 15.5 days (with standard deviation of 6.1 days) per participant. Of course, the phone usage patterns vary a lot between subjects, and the number of events for some

participants was low even on full days of phone data available, making their data sparse.

In previous studies, it was found that the answer rate to the pop-up questionnaires was less than half and they were filled in at random times. In our study, the subjects had 30 minutes to answer the questionnaire or it was closed. Due to application malfunction, the information regarding unanswered questionnaires was not always transmitted to the server. Because of this, on average 91% of the questionnaires per participant showed up in the data, with 78% of them answered. If we consider all the questionnaires (3 times the number of days available), the average answer rate was 71% which is still higher than in previous studies. This may be because we had fewer questionnaires per day, and so the participants did not get tired from answering them.

Figure 11 shows one week of data from one participant as a heatmap where each value is the mean over one-hour windows. All the data is visualised before doing the splitting to night-time and daytime data. The day label on x-axis is placed at midnight. The first block from upwards concerns location, the second questionnaire answers, the third phone usage variables and the last physiological variables. The descriptions of the features are presented in Section 3.2.3.

Location variables show a clear routine where nights and weekend is mostly spent at location 2, and location 3 is visited for several hours almost each business day but not on weekend. On the days when location 3 is visited, the middle (4 pm) questionnaire for feeling of control shows lighter values than at other times, meaning that the person has felt less in control. The person has not used applications from categories shopping and other at all, and it seems that most common application categories for him during the week are communication, infotainment, utility, and unknown. Based on screen on time (scrn_on), it appears that phone was used less at times when the person was at location 3 although it is hard to say for certain from this kind of presentation.

None of the physiological variables show clear variation which may be an issue regarding data quality or the simple fact that we are visualising hourly means. However, the bottom variable (valid_ibis) describing the amount of valid IBI mea-

Figure 11: One week of feature-level real-life data depicted as means over one-hour windows. A darker shade means a higher value, using feature-wise minmax-normalisation, and pure white means missing data. The time span for pop-up questionnaire answers was lengthened to make the answers visible.

surements clearly shows that most of the valid samples are seen around and after midnight.

3.2.2 Assessing Stress

Having three questionnaires a day makes it difficult to assign labels to an exact point in time, other than the minute the questions were answered at. The answers reflect the persons' current feelings which may vary drastically. Imagine that a person answers that he is calm and totally in control and five minutes later his manager comes to tell him that he must give a presentation to a customer the same afternoon. Both aspects (calmness and feeling of control) would probably fall to the other end of the scale. As the previous studies discussed in the Introduction indicate, minute-by-minute stress detection is hard even for supervised systems. A recent example of this is the study by (Smets et al., 2018) who found an F1-score of 0.43 which is little better than assigning all the samples to the non-stressed class.

Moreover, our method will not find a stress and a non-stress cluster but for this kind of data it finds *behaviour patterns*. A little similar to what was done in (Vildjiounaite et al., 2017, 2018), we identify normal daily behaviour and assume that stress is a type of abnormal behaviour. At night it may be realised as e.g. elevated HR or HRV. During the day abnormal phone usage may indicate stress. If, for example, a person feels overpowered by the stressful situation he may just browse his phone or if the same person feels that he may overcome the situation by working extremely hard, he may not use his phone at all.

Because of these difficulties we only attempt on detecting stressful and non-stressful days instead of more fine-grained classification. This should be enough to assess the level of episodic acute and chronic stress which are the two harmful types of stress, and thus daily level detection does not introduce a limitation to model usability.

We relate the found behaviour patterns (i.e. cluster labels) to stress by calculating *dayscores* that reflect how much the daily behaviour differs from normal. We determine "normal behaviour" by first dividing each day (or night) into w

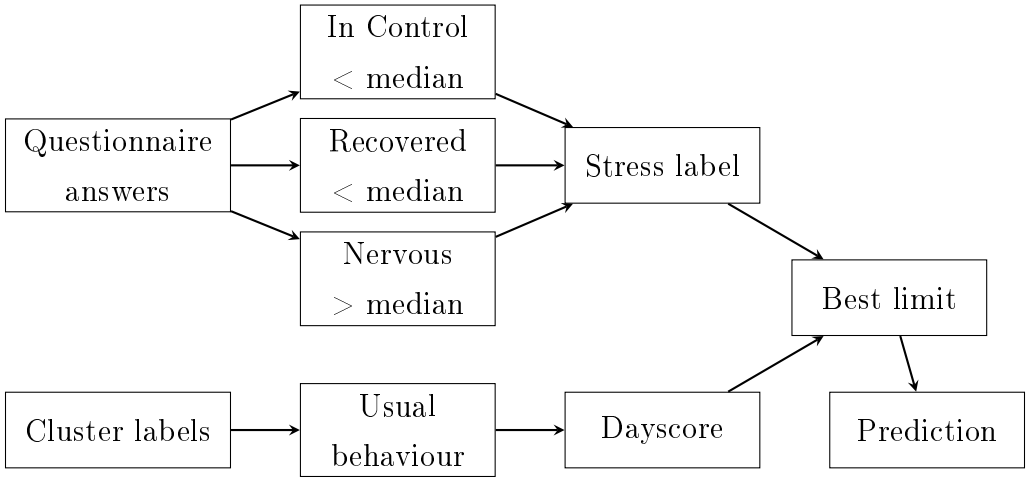


Figure 12: The inference procedure for the real-life data. Stress label gets the value one if at least two of the three preceding conditions hold. The best limit is usually found using only the training data.

time windows. For each given window we find normal behaviour by two methods called *simple* and *conditioned*. The simple version is the mode of cluster label for each window: the usual behaviour at window i is the most common cluster label observed during the window, across all the days. The conditioned version is the same but it takes context into account: the usual behaviour at window i is the most common cluster label given weekday and location, across all days. Weekday was used as a binary variable indicating either business day or weekend.

Since routines may differ a lot even for two persons let alone a group of people, determining usual behaviour in a general way is unjustifiable. Therefore, we always determine usual behaviour in a person-specific fashion regardless of the SOM personalisation level, and so even the most general version of SOM now contains a person-specific element. This also means that a totally general prediction is not considered.

The dayscore is calculated by incurring a penalty for deviations from usual behaviour. Assume there are $c \in \mathbb{N}$ clusters found for a given SOM and let $\boldsymbol{\mu}_k$ denote the mean vector of all the SOM prototypes in cluster k. Further, let k_{obs} be the observed cluster label and k_{us} the usual cluster label at given window j. We

consider the ordered sequence of distances

$$(d_i^*)_{i=0}^{c-1} \quad \text{with } d_i^* \leq d_{i'}^* \text{ for } i < i',$$

where $d_i^* = d(\boldsymbol{\mu}_{k_{obs}}, \boldsymbol{\mu}_i)$ and d is the distance used in training SOM. The window score $w_s^{(j)}$ at window j is the index m at which $d_m^* = d_{k_{us}}^*$ and the dayscore is given by

$$\text{DS} = \sum_{j=1}^{w} w_s^{(j)}. \tag{3.2}$$

By this definition, the window score equals to zero when the observed behaviour is the same as usual behaviour (if $k_{obs} = k_{us}$, then $d_{k_{obs}}^* = d_{k_{us}}^* = 0$ and so $m = 0$) and gets higher scores the more actual behaviour differs from normal behaviour. The idea for using dayscores in daily stress assessment comes from (Vildjiounaite et al., 2017, 2018) but the way the scores are calculated here was ideated by the author.

The dayscores are binarised using the function

$$\delta(\text{DS}, \lambda) = \begin{cases} 0, & \text{DS} \leq \lambda \\ 1, & \text{DS} > \lambda, \end{cases} \tag{3.3}$$

where λ is a given limit. The optimal limit is found by conducting a grid search where we maximise the F1-score between the training data ground truth labels and binarised dayscores. The tested values for λ are taken between $\mu_{\text{DSS}} - 1.5 \cdot \sigma_{\text{DSS}}$ and $\mu_{\text{DSS}} + 1.5 \cdot \sigma_{\text{DSS}}$ where DSS denotes the training data dayscores, μ their mean and σ their standard deviation. Determining the optimal limit in this fashion adds a certain amount of supervision but if we used a hard limit given as a parameter, the process would be completely unsupervised.

As a ground truth, we use a person-specific combination of the pop-up questionnaire answers. A day is labelled as stress if at least two simultaneous answers to the feeling of control, recovery, and nervousness are more negative than is usual for the person and this must happen twice during the day. That is, if a person feels two of less in control, less recovered or more nervous than their median answer to that question twice during the day, the day is labelled as stress. We acknowledge that this may not necessarily mean true stress but it certainly indicates that

something unusual and unsettling has gone on. A high-level flowchart of the whole stress assessment procedure is shown in Figure 12.

3.2.3 Data Preprocessing

Checking the data quality was in truth the first step of data preprocessing. As mentioned, the night-time physiological measurements of 35 subjects and the phone data of 65 subjects could be used. Phone data at night would mostly contain nothing (people do not use their phones when they are asleep) and so we used only daytime (6 am to 12 pm) phone data.

Unlike with WESAD data, we calculated some basic features previously used in literature. As is customary (e.g. (Huysmans et al., 2018; Vildjiounaite et al., 2017, 2018)), we employed overlapping windows. For the physiological features, we selected a conventional 5 minute calculation period (Shaffer and Ginsberg, 2017) with 4 minute overlap to obtain a minute-by-minute data.

To catch phone usage behaviour, longer windows are needed because people do not necessarily interact with their phones regularly, especially when working. Previously, (Vildjiounaite et al., 2018) used three-hour windows with one hour overlap. Following that, we performed an experiment on window lengths between one and four hours with $25-50\%$ overlap. We calculated personal SOMs and

Table 7: List of extracted features for the real-life data.

Data	Feature Name
Physiological Data	Mean interbeat interval
	Mean heart rate
	RMSSD
	ACC_{SD}
Phone Usage Data	Mean screen on time
	Number of screen state changes
	Category-wise mean application usage

clustered them with the clustering method and parameters determined by the silhouette score. Because two found clusters would just identify whether the phone was used or not, we based the decision also on the number of clusters found, preferring a higher number. After this, the window size for phone features was set to one hour with thirty-minute overlap.

For the physiological data, we calculated mean heart rate, mean IBI, RMSSD and the standard deviation of the magnitude of acceleration ($\mathrm{ACC_{SD}}$). All the features were calculated over all the available sensor values within that window. The first two are simple means, RMSSD is given by

$$\mathrm{RMSSD} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n-1}(\mathrm{IBI}_{i+1} - \mathrm{IBI}_i)^2} \tag{3.4}$$

and $\mathrm{ACC_{SD}}$ by

$$\mathrm{ACC_{SD}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\mathrm{ACC}_i - \overline{\mathrm{ACC}})}, \quad \text{where} \tag{3.5}$$

$$\mathrm{ACC} = \sqrt{\mathrm{ACC}_x^2 + \mathrm{ACC}_y^2 + \mathrm{ACC}_z^2}, \tag{3.6}$$

i.e. the magnitude of acceleration is the Euclidean norm of its components. These features have previously been used by e.g. (Smets et al., 2018).

During data collection, we simply stored the timestamp at which an application came to the foreground and went to the background (or was replaced by another app in the foreground). Similarly, we stored the timestamps when the phone screen was turned on or off. In the preprocessing stage, we first calculated a minute-by-minute data of how long each application category was in the foreground and screen was turned on. Then we calculated the mean of how long per minute each category was active or screen was on during each window. In addition, we calculated the number of screen state changes. The features were calculated in varying window sizes, as explained above, and we finally settled to one-hour windows with thirty minutes overlap. Similar phone usage features have previously been used in (Ciman and Wac, 2018) and (Vildjiounaite et al., 2017, 2018). All the features extracted are summarised in Table 7.

3.2.4 Results

Like in the analysis of the WESAD data, we fit SOMs in personal, semi-personal and general levels and we used the minmax-normalisation (Eq. 3.1) personallu and generally. For the semi-personal models the subjects were clustered with K-Means in a similar manner as explained in Section 3.1.1 and we ended up with three user-clusters for daytime data and four user-clusters for the night-time data. Again, we used leave-one-subject-out cross-validation for the semi-personal and general model. For the personal models we used leave-one-day-out cross-validation which means that each day in turn was left for testing and the rest were used for training.

For the daytime phone usage data, we considered the ground truth described in Section 3.2.2. For the night data, it is not clear should we use the label of the previous day or the same day: a stressed day may show up somehow in the following night data, or a person may feel stressed because of poorly slept night. Thus we considered both the previous and the same day labels for the night data.

To allow for further assessment of the effect of personalisation, the best limit in Eq. (3.3) was determined in a general and a personal way for the semi-personal and general models. In the general version, the limit is found so as to maximise training data F1-score. In the personal version, we use leave-one-day-out validation to maximise testing data F1-score. For the conditioned usual behaviour in the personal version, it sometimes happened that a condition (a window, weekday and location combination) in test data was not available in the training data. This situation is definitely abnormal and we added plus one to the dayscore in these cases.

The SOM topology was set in a similar fashion as explained in Section 3.1.2. After the SOMs were calculated and the best clustering for each SOM was found, we evaluated the dayscores. Usual behaviour was always determined separately for each person in half an hour windows. Since we used one-hour windows with half an hour overlap for the daytime data, it actually covered the time frame 5.30 am to 12 pm instead of 6 am to 12 pm and so the number of windows was 37. For the night-time data, there were 12 windows (12 am to 6 am in half an hour intervals). The limit λ in Eq. (3.3) was determined by maximising the F1-score with training

data (general) or with LODO-validation (personal).

Now the amount of data was so high that running the analysis cycle multiple times was not feasible for this study. Therefore, the values reported are means and standard deviations across SOMs and not across SOMs *and* runs as previously.

3.2.4.1 Clustering results

The clustering results for the real-life data are shown in Table 8. As is expected, the amount of noise in the data is higher now and this time HDBSCAN* clusters at least 80% of the data for three SOMs at the personal level, however, for the three SOMs it clustered it was always found to give the highest silhouette score. Often it refused to cluster any of the data and so we have left out the corresponding rows in Table 8. Focusing on the other two methods, the results seem more consistent across different personalisation levels than was the case with WESAD data.

For the night-time data GMM and K-Means always find on average $2-3$ clusters at all the personalisation levels with generally small standard deviation. The silhouette scores range mostly around $0.30-0.45$ with a low deviation. Similarly to WESAD data, we note that personal normalisation does not seem to have a positive effect on the data clusterability at semi-personal and general levels. Unlike with WESAD data, the number of clusters found and the silhouette score values at the most general level are almost the same as on the personal level. All this tells us that during the night there are no major differences between individuals' behaviour, and similar patterns are found regardless of the personalisation level.

This is not the case for the daytime data. At the personal level, there are approximately $3-4$ clusters found (with a high standard deviation) and the silhouette score is around 0.50. As the level of personalisation decreases, so does the number of clusters found but the silhouette score tends to stay the same or increase. This indicates that the number of different phone usage patterns found decreases as the level of generalisation grows: more general SOMs find more general patterns. Some detailed patterns found on the personal level may actually be part of some bigger pattern found when looking at the data on a more general level and so the different patterns are combined.

Table 8: Clustering results for the real-life data. The number of SOMs at personal level was 1008 at day and 441 at night, and 65 (day) and (35) for the other conditions. The values are means with standard deviation in parenthesis (Clusters, Silhouette, Clust (%)), and proportion of times chosen as the best method (Best (%)) and available (Av (%)).

Model	Daytime	Method	Clusters	Silhouette	Clust (%)	Best (%)	Av (%)
PERS	night	GMM	2.6 (1.2)	0.31 (0.09)	100	3.6	
		K-M	2.5 (0.8)	0.47 (0.05)	100	96.4	
		HDBS	0.5 (0.9)	0.15 (0.27)	8 (16)	0	
	day	GMM	4.1 (2.6)	0.48 (0.12)	100	12.4	
		K-M	3.4 (2.2)	0.55 (0.09)	100	87.3	
		HDBS	0 (0.3)	0.02 (0.11)	1 (10)	0	
		$HDBS_{\geq 0.8}$	2.0 (0)	0.74 (0.08)	84 (1)	0.3	0.3
PSPN	night	GMM	2.8 (1.4)	0.17 (0.04)	100	0	
		K-M	2.9 (0.3)	0.37 (0.02)	100	100	
		HDBS	0.2 (0.6)	0.05 (0.18)	2 (7)	0	
	day	GMM	2.2 (1.0)	0.43 (0.04)	100	0	
		K-M	2.2 (0.5)	0.52 (0.03)	100	100	
PSCN	night	GMM	2.9 (1.0)	0.27 (0.09)	100	0	
		K-M	2.4 (0.5)	0.45 (0.02)	100	100	
		HDBS	0.1 (0.3)	0.02 (0.10)	1 (3)	0	
	day	GMM	2.0 (0.1)	0.52 (0.06)	100	1.5	
		K-M	3.7 (1.6)	0.68 (0.05)	100	98.5	
GPN	night	GMM	2.1 (0.2)	0.14 (0.02)	100	0	
		K-M	3.0 (0.2)	0.36 (0)	100	100	
	day	GMM	2.0 (0.2)	0.41 (0.03)	100	0	
		K-M	2.0 (0)	0.52 (0.01)	100	100	
GGN	night	GMM	2.4 (0.6)	0.30 (0.04)	100	0	
		K-M	2.9 (0.4)	0.44 (0.01)	100	100	
	day	GMM	2.0 (0)	0.53 (0.02)	100	0	
		K-M	3.0 (0.4)	0.68 (0.01)	100	100	

PERS: personal, PSPN/PSCN: person-similarity with personal/cluster-wise normalisation, GPN/GGN: general with personal/general normalisation, K-M: KMeans, HDBS: HDBSCAN*, $HDBS_{\geq 0.8}$: HDBSCAN* when it clustered at least 80% of the data.

K-Means was the most often chosen clustering method by far, with GMM being chosen at two out of five personalisation levels and even then not too often. This strengthens the comment in Section 3.1.3 that estimating the covariance structure of the SOM prototypes does not improve clustering performance.

3.2.4.2 Prediction results

Because of lack of answers to pop-up questionnaires, we could not evaluate the prediction results for all the users and all the days across all the conditions and so the number of users and days available for each condition are reported. We report the F1-score, taken as the weighted average of each participants individual F1-score (weighted with the number of days per subject) and as a global measure with instances from all the participants.

We used Markov Chain Monte Carlo -simulation to obtain an estimate of a baseline F1-score. In total out of the 1008 days available, 441 (43.75 %) were reported as stress. For each day, we randomly sampled a number from Bernoulli's distribution with $p = 0.4375$ and then calculated the F1-score of randomly sampled prediction and the given label. Mean F1-score and its standard deviation were observed to stabilize after a few hundred iterations and so after 5000 MCMC-iterations and a burn-in period of 500, the mean F1-score was equal to 0.438. If we do not use the prior information on label distribution and choose $p = 0.5$, the same procedure yields mean F1-score of 0.466.

The daytime stress prediction scores based on phone usage data are shown in Table 9. The global F1-score seems to give slightly higher scores than the weighted average but usually the difference is not high. The standard deviation of the weighted average is always high which tells us that there are big differences between individuals in prediction performance. There are no big differences between the different levels of personalisation of SOMs but choosing the limit λ in Eq. (3.3) in a personalised way seems to produce higher scores. The conditioned version of determining usual behaviour seems to perform better in almost all the conditions, telling that behaviour depends on context and it should be taken into account when building the prediction model. In addition, all the scores are above

random guessing although not by much.

Table 9: Prediction results for the daytime data. At daytime, we predict stress for the current day. We could do the prediction for 58 participants and 930 days at the personal level and 65 participants and 1008 days at the other levels. F1-global is the F1-score over all the available days. F1-average is the weighted mean of subject-by-subject F1-scores with standard deviation in parentheses, weighted with the number days per subject.

Model	Limit	Usual behaviour	F1-global	F1-average
PERS	personal	simple	0.57	0.53 (0.19)
		conditioned	0.60	0.57 (0.17)
PSPN	personal	simple	0.59	0.57 (0.20)
		conditioned	0.62	0.60 (0.21)
	general	simple	0.52	0.48 (0.22)
		conditioned	0.51	0.47 (0.21)
PSCN	personal	simple	0.58	0.55 (0.21)
		conditioned	0.63	0.60 (0.21)
	general	simple	0.50	0.44 (0.24)
		conditioned	0.52	0.45 (0.24)
GPN	personal	simple	0.58	0.55 (0.20)
		conditioned	0.61	0.59 (0.20)
	general	simple	0.52	0.47 (0.23)
		conditioned	0.51	0.45 (0.24)
GGN	personal	simple	0.57	0.54 (0.21)
		conditioned	0.63	0.60 (0.21)
	general	simple	0.51	0.44 (0.24)
		conditioned	0.52	0.47 (0.22)

PERS: personal, PSPN/PSCN: person-similarity with personal/cluster-wise normalisation, GPN/GGN: general with personal/general normalisation.

The stress prediction scores with the night-time data are shown in Table 10. Firstly we note that also here the global F1-score is higher than the weighted average of participant scores. The standard deviations of the weighted averages are even higher than we saw with the daytime data and we make the same conclusion that individual differences are high. In general, it seems that the scores here are lower than the ones with daytime data, especially if we look at the weighted average scores, and many of the scores seem to be even lower than random guessing. This may be a symptom caused by data quality problems because nights with a low amount of data would always obtain low dayscores.

The scores for predicting the following day stress are perhaps a little higher than those predicting the previous day stress but the differences are small. Here, too, we see that choosing the limit in dayscore binarisation in a personalised way seems to give better scores than the general version and that the conditioned version of usual behaviour identification seems to be better. If we consider the personalisation level of SOMs, there are no big differences but the highest values and smallest standard deviations are seen on the personal level.

If we want to build an unobtrusive stress detection system, ideally the system users would not need to fill in any questionnaires. Setting a personal dayscore binarisation limit reduces model generalisability because we could not do prediction until sufficient amount of questionnaire answers are obtained for each person. Still, we would need to obtain enough data to determine the usual behaviour. Having this in mind the following analyses are conducted with the most general version available, general SOM with general normalisation and binarisation limit but using the conditioned usual behaviour detection.

3.2.4.3 Describing Behaviour Patterns and Cluster Contains

To interpret the results and to inspect what constitutes stressful behaviour, we shortly take a look at how stressful and non-stressful days differ and what do the found behaviour patterns mean. Because the prediction scores for daytime data were higher than for night-time data, for this task we used only the daytime data and calculated a new, general SOM using all the participants' data in training.

Table 10: Prediction results with night-time data for the following (same) and previous day. F1-global is the score over all days and and F1-average is the weighted-by-number-of-days subject-by-subject mean with standard deviation in parentheses.

Model	Day	Limit	Usual behaviour	Users	Days	F1-global	F1-average
PERS	same	personal	simple	26	338	0.54	0.51 (0.22)
			conditioned	26	338	0.55	0.52 (0.19)
	previous	personal	simple	27	344	0.54	0.51 (0.16)
			conditioned	27	344	0.59	0.57 (0.13)
PSPN	same	personal	simple	34	415	0.51	0.48 (0.27)
			conditioned	34	415	0.53	0.50 (0.25)
		general	simple	34	415	0.47	0.40 (0.27)
			conditioned	34	415	0.43	0.36 (0.26)
	previous	personal	simple	35	422	0.50	0.48 (0.22)
			conditioned	35	422	0.50	0.48 (0.23)
		general	simple	35	422	0.41	0.36 (0.23)
			conditioned	35	422	0.40	0.35 (0.24)
PSCN	same	personal	simple	34	415	0.51	0.48 (0.27)
			conditioned	34	415	0.54	0.51 (0.27)
		general	simple	34	415	0.48	0.41 (0.27)
			conditioned	34	415	0.43	0.36 (0.27)
	previous	personal	simple	35	422	0.50	0.47 (0.22)
			conditioned	35	422	0.53	0.51 (0.23)
		general	simple	35	422	0.43	0.38 (0.23)
			conditioned	35	422	0.40	0.34 (0.26)
GPN	same	personal	simple	34	415	0.52	0.49 (0.27)
			conditioned	34	415	0.53	0.51 (0.25)
		general	simple	34	415	0.44	0.38 (0.26)
			conditioned	34	415	0.42	0.35 (0.25)
	previous	personal	simple	35	422	0.50	0.48 (0.22)
			conditioned	35	422	0.53	0.52 (0.23)
		general	simple	35	422	0.40	0.35 (0.23)
			conditioned	35	422	0.43	0.37 (0.24)
GGN	same	personal	simple	34	415	0.52	0.48 (0.27)
			conditioned	34	415	0.52	0.50 (0.26)
		general	simple	34	415	0.43	0.37 (0.25)
			conditioned	34	415	0.48	0.40 (0.25)
	previous	personal	simple	35	422	0.50	0.48 (0.22)
			conditioned	35	422	0.52	0.51 (0.23)
		general	simple	35	422	0.41	0.35 (0.24)
			conditioned	35	422	0.47	0.41 (0.25)

PERS: personal, PSPN/PSCN: person-similarity with personal/cluster-wise normalisation, GPN/GGN: general with personal/general normalisation.

The cluster-wise mean feature values for the calculated SOM with all the participants' data are shown in Figure 13. Three clusters were found. Based on this image we interpret the cluster with label 2 as high phone usage cluster and the cluster 0 as the low usage cluster. The cluster with label 1 contains nearly all the times communication applications were used. Thus the phone usage behaviour can broadly be categorised as no usage, general usage, and communication.

During high usage, the phone screen is on around 40 seconds each minute and most of the application categories have been used at least a few seconds, on average, and the number of screen state changes is the highest, though not by much. The amount of screen state changes for the low usage cluster is almost the same as for the other two, perhaps indicating that the phone was mostly used for checking the time. Overall the application category "communication" was by far the most used and it is no surprise that it is so indicative that it constitutes a cluster of its own.

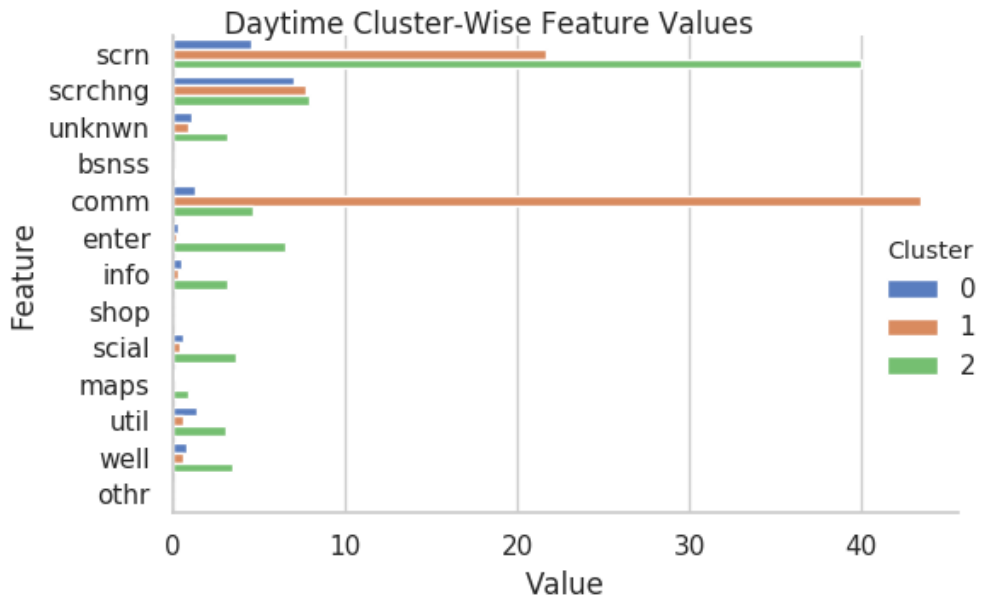


Figure 13: The cluster-wise means of daytime feature values. The labels are abbreviations of screen on time, number of screen changes and each application category. The value for screen changes is the average amount over daytime windows and for other features it is the average seconds used per minute over daytime windows.

To compare between stressful and non-stressful days, mean feature values are shown in Figure 14. The daily predicted stress was obtained from the general leave-one-user-out SOMs (corresponds to GGN - general - conditioned in Table 9). The most notable differences are in screen on time and the number of screen state changes. On stressful days the screen state is changed about one more time per each window but also the screen on time is about one second higher per minute than on non-stressful days. The former refers to more erratic behaviour and because screen usually stays on at least for a few seconds, it may be the reason for the latter.

Similarly to (Huysmans et al., 2018), we found the U-matrix inspection inconclusive. At each level of SOM personalisation the U-matrices were found to be similar to those shown in Figure 9, but showing no clear borders even for personal SOMs. This probably means that differences between different behaviour patterns are not large enough to be captured in the U-matrix, or that the data are not clusterable at all and there are no true patterns to be found. The latter conclusion

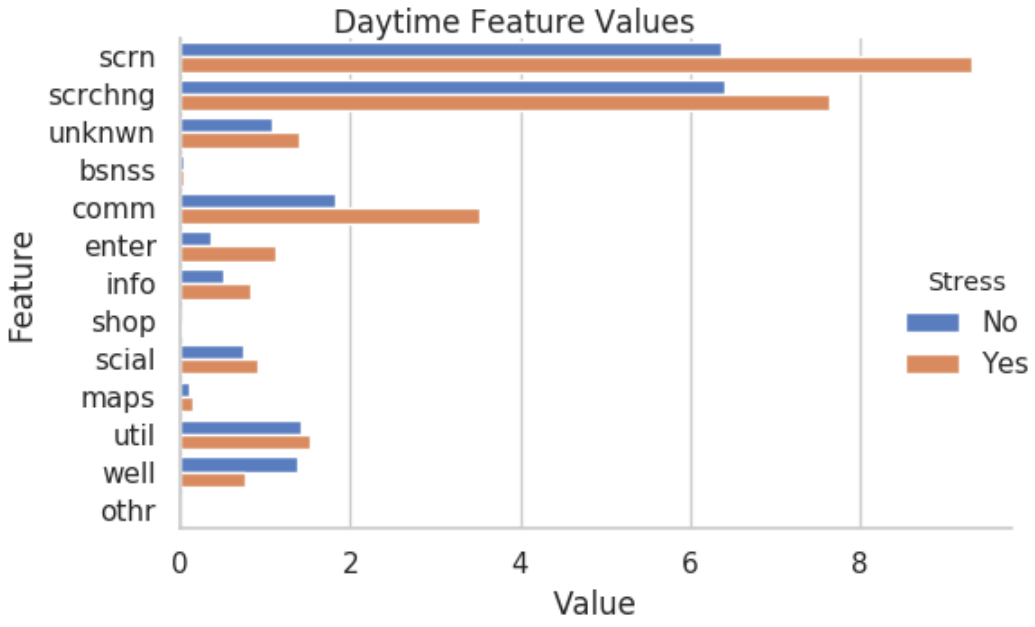


Figure 14: The means of daytime feature values for stressed and non-stressed days. The labels and interpretation of values are the same as in Figure 13.

Table 11: The Pearson's chi-squared test scores for the behaviour differences between stressful and non-stressful days, both for reported and predicted labels. The columns cl0 - cl2 denote the number and the proportion of time windows with the corresponding label. The column df denotes degrees of freedom and χ^2 the value of the test statistic. The total amount of time windows was $n = 37296$.

Label	Stress	cl0	cl1	cl2	df	χ^2	P-value
Reported	no stress	18582	654	1743			
		0.886	0.031	0.083			
	stress	14583	419	1315			
		0.894	0.026	0.081	2	11	0.004
Predicted	no stress	15885	236	1010			
		0.927	0.014	0.059			
	stress	17280	837	2048			
		0.857	0.042	0.102	2	504	< 0.001

is given credit by the fact that HDBSCAN* clustered nearly none of the data. On the other hand, the silhouette score for the calculated general SOM with forced clustering was 0.68, denoting a clustering with high cohesion and separation.

To find out whether behaviour differed between stressed and non-stressed days, we applied Pearson's chi-square test to determine the statistical significance of the differences. We described a day by a sequence of cluster labels (behaviour patterns) and simply compared whether the distribution of labels was different, both for reported and predicted labels.

The test scores are summarised in Table 11. Both the predicted and reported version show a statistically significant difference in behaviour. For easier comparison, we have presented the condition-wise relative amounts of each type of behaviour. The differences are small for the reported label values but they show more of low phone usage behaviour and less of other types of behaviour. The differences are larger for the predicted labels and they show less of low phone usage

and more of the other two types, especially high usage behaviour.

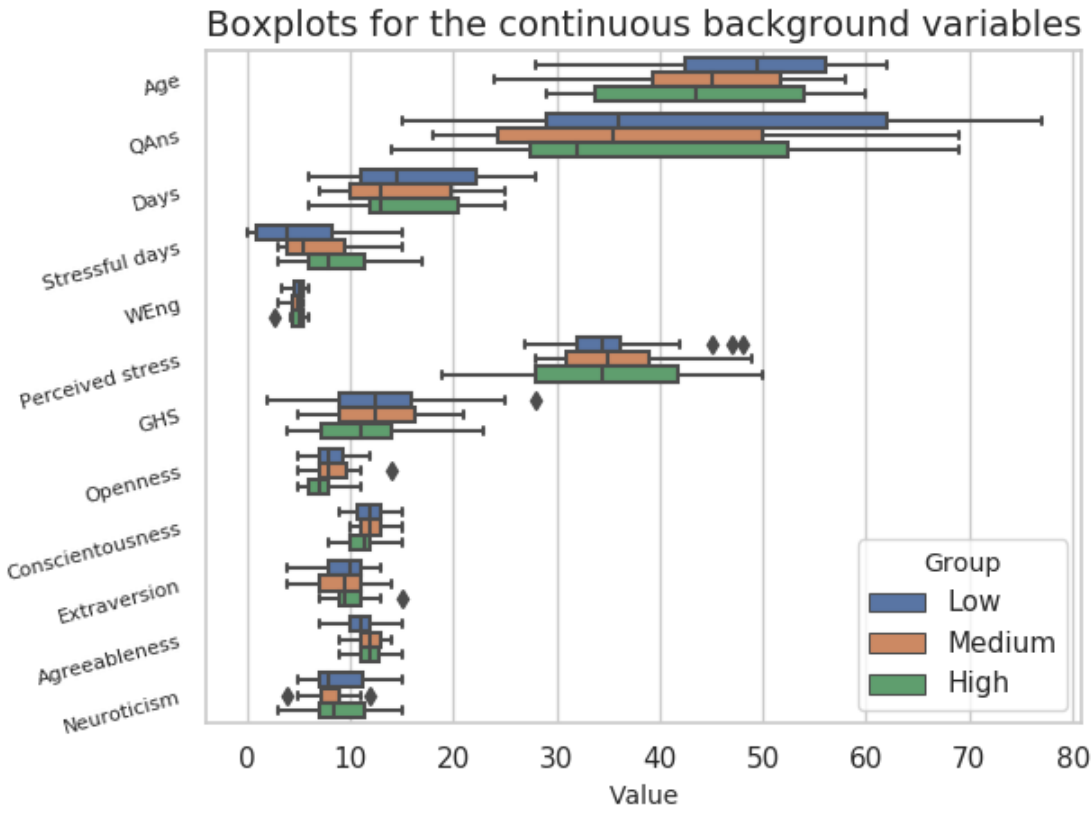
These results are conflicting because one would expect the differences to be to the same direction. Because behaviour is mostly from the low phone usage cluster the predicted label (that describes abnormal behaviour) shows stress for the days with more phone usage. However, the reported stress describes abnormal feelings that do not seem to reflect on phone usage as clearly.

3.2.4.4 Associations to Background Variables

As noted earlier, the prediction scores differ a lot between users. The reason for this may lie in different personality traits or other demographics of the subjects or simply that some users do not use their phones enough to allow for reasonable behaviour pattern detection. To estimate these effects, we divided the participants according to prediction scores found with the most general model available into groups of low performance ($F1 < 0.33, n_1 = 15$), medium performance ($0.33 < F1 < 0.66, n_2 = 31$) and high performance ($F1 > 0.66, n_3 = 19$) and compared the values of several background variables within those groups. Because one person (from the medium performance group) did not answer the prequestionnaire, most of the tests had a total $n = 64$.

Previously, a similar inspection was conducted by (Smets et al., 2018) and they found that high performers had more imbalanced self-reports, a healthier lifestyle and lower depression, anxiety and stress levels than the subjects in the low performing group, and they were older. They found no difference between genders. Similarly, we inspect the effects of gender, age, personality, health, and reported phone usage.

The distributions of tested variables are shown in Figure 15. Mostly it seems that there are no major differences between performance groups in any of the visualised variables. The most notable ones are in variables "Stressful days" where the median of a lower performing group (low vs. medium and medium vs. high) is around the lower quartile of the higher performing group, and "Perceived stress" where the range and the interquartile range of the high performance group are wider than for the other two groups.



QAns: number of pop-up questionnaires answered, WEng: work engagement, GHS: general health state.

Figure 15: The distributions of continuous background variables within performance groups.

To check the statistical significance of the differences between performance groups we employed the Kruskal-Wallis test for continuous variables and the Pearson's chi-squared test for categorical variables. The Kruskal-Wallis test was chosen over ANOVA because none of the continuous background variables were normally distributed. The null hypothesis for the Pearson's test is that the distribution of low, medium and high performers is independent along the levels of the background variable. The null hypothesis for the Kruskal-Wallis test is that the distribution of the background variable is the same in all the performance groups.

The results for the test conducted are presented in Table 12. For continuous

Table 12: The prediction score associations with background variables. The values for continuous variables are means with standard deviation in parenthesis. For catecorigal variables the number of participants in each category is shown.

Background Variable	Feature Value			Test Statistic	P-value
	Group				
	Low	Medium	High		
Age	48.7 (7.1)	44.9 (10.7)	44.1 (10.5)	H = 1.146	0.563
Gender	F: 8, M: 7	F: 22, M: 9	F 10, M: 8	χ^2 = 1.857	0.395
Days available	16 (6)	16 (7)	15 (6)	H = 0.117	0.943
QAns	40 (16)	41 (19)	39 (16)	H = 0.066	0.967
Stressful days	4.7 (5.1)	6.6 (3.8)	8.8 (4.0)	H = 8.701	0.013
WEng	5.0 (0.8)	4.8 (0.7)	5.0 (0.8)	H = 1.187	0.552
GHS	14.3 (7)	12.4 (4.2)	11.6 (5.0)	H = 1.431	0.489
Perceived stress	36.7 (5.8)	34.4 (5.8)	34.8 (8.9	H = 1.533	0.465
Openness	7.9 (1.4)	8.6 (2.1)	7.4 (1.7)	H = 4.178	0.124
Conscientiousness	11.9 (1.8)	12.1 (1.4)	11.3 (2.0)	H = 2.735	0.255
Extraversion	9.7 (2.4)	9.1 (2.6)	10.2 (2.1)	H = 1.325	0.516
Agreeableness	10.6 (2.2)	11.7 (1.7)	11.9 (1.7)	H = 4.060	0.131
Neuroticism	8.7 (2.8)	8.6 (2.3)	8.8 (3.2)	H = 0.284	0.868
Phone Usage	H: 7, L: 8	H: 14, L: 17	H: 13, L: 6	χ^2 = 2.803	0.246

QAns: number of pop-up questionnaires answered, WEng: work engagement, GHS: general health state.

For gender, F = Female and M = Male and for phone usage H = High and L = Low.

background variables we show the mean and standard deviation by performance group, the Kruskal-Wallis test statistic H and the corresponding P-value. Similarly for categorical variables we show the distribution within each group, the Pearson's chi-squared test statictic χ^2 and the corresponding P-value.

On average there was no difference in the amount of pop-up questionnaires answered or in the amount of days available but there were more reported stressful days in higher performing groups. There was no difference in gender or age.

Similarly we found no significant differences for feeling of work engagement (Hakanen, 2009) (reference value 4.4), general health state (Gnambs and Staufenbiel, 2018) (reference value 10.3 with standard deviation 5.0, higher scores are worse) or perceived stress (Cohen et al., 1983) (reference value 23.2, higher scores are worse).

In the pre-questionnaire, the subjects also filled in a Big Five personality trait test (Soto and John, 2017) to determine their level of openness, conscientiousness, extraversion, agreeableness, and neuroticism. We tested the effect of personality as a continuous variable (score for each trait) but all the tests showed no significant difference.

Lastly, we checked the effect of reported phone usage. We asked whether the subjects always carry the phone with them at work and at leisure and whether they use the phone applications at work and at leisure. The participants who answered "Yes" to all of the four questions were labelled with high phone usage and the rest with low phone usage. We found no significant difference in prediction score in the high usage group versus the low usage group.

Based on the tests conducted here the only statistically significant aspect was the amount of reported stress in the pop-up questionnaires. The high-performance group reported 56% of days as stressful, the medium performance group reported 43% and the low-performance group reported 27%. However, the values for predicted stress are, respectively, 75%, 54% and 46%. A similar difference was found by (Smets et al., 2018) but their high-performance group reported less stress than the low-performance group.

4 Discussion and Future Work

We have presented a personalised method based on SOM and clustering for mental stress detection. The method used is fully unsupervised up until the point the cluster labels are related to stress labels but we also commented on how to make it totally unsupervised. In stress detection context, this was the first time when different personalisation options combined with an unsupervised method have been compared to this extent, the first time an unsupervised method was applied to both laboratory and real-life datasets and the first time when multiple clustering options for SOM-based models were considered.

As such, the results are promising. The laboratory data results show that SOM can indeed detect different responses in multi-dimensional behavioural data, and while personalisation does not improve the clustering results it does improve the prediction scores. The personalised models performed better also for the real-life data even though the differences were not as clear.

The best prediction scores for the laboratory data were obtained with the fully personal model and the general model with personal feature normalisation. With the real-life data, the differences between the SOM personalisation levels were small which may be due to determining usual behaviour individually for each participant. It may be that general behaviour patterns are able to describe each person's behaviour sufficiently and individual differences are caught with the usual behaviour detection. Another aspect is that setting a personal limit for the dayscore binarisation clearly improved the prediction scores but using that approach requires system users to fill in a lot of questionnaires before obtaining stress predictions.

We did not find a supreme solution for personalisation and it remains an open question at what level and how it should be conducted. The performance with fully personalised models was always high but training the models separately for each person and especially drawing conclusions on how they perform the predictions is troublesome and requires much manual work. In this study, we only investigated what are the predictions for real-life daytime data based on but this is something

that must be done before any stress detection system can be put to practice. If there are separate models for each system user there are as many investigations to be done as there are system users and therefore a more general version should be favoured.

Another option to personalisation could be using *transfer learning* which means that knowledge learned by solving a problem is utilised to solving another problem. Because it is always possible to continue training SOM with a new set of data, we could first calculate a "baseline" SOM for a group of people and then continue training with data from each participant. We did not think we had enough good quality data for validating the results for each participant and so this approach was not considered and is left for future research.

The biggest differences between laboratory and real-life data lie in data quality and validation of results. In the laboratory, it is easy to obtain a lot of high quality, labelled data and it is easy to draw conclusions from the data with high confidence. Obtaining the data in real-life is subject to sensor malfunctions, improper attachment and participants tiring up to the data collection. There is a need for more reliable and more unobtrusive wearable sensors to collect trustworthy physiological data continuously, and for a method for validating the behaviour patterns found.

In our study, the quality of physiological measurements was so low that the night-time data for about only half of the participants could be used. We did not attempt on combining the night and daytime measurement predictions but this is something that should be done in the future. Moreover, if reliable physiological data were available for daytime, combining that with phone usage data might reveal patterns not found otherwise. For fixing the quality of the existing data, methods like multiple imputation to account for missing data could be investigated, or the knowledge of how much data are missing and when could be utilised in stress prediction directly.

The ground truth label we extracted for the real-life data described abnormal conditions more than stress. To obtain the predicted label, we assumed that stress is an abnormal type of behaviour. For the most dangerous case when the person is always stressed, this method cannot find any stressed days for him. More-

over, the results highlight that reported abnormal feelings do not necessarily show up in phone data when they happen, partly because we do not use our phones constantly. The feelings may, however, show up in physiological data and so this further underlines the need for continuous physiological measurement.

However, our results presented should be treated as observational and their generalisability is questionable. The sample sizes were quite small (15 for laboratory data, 65 and 35 for the daytime and night-time real-life data, respectively) and the duration of data collection for the real-life data was quite short (four weeks but approximately two weeks per participant could be used). In addition, both of the samples were homogeneous regarding working status, age and sex. Future studies should focus on longer data collection periods with more participants from a more heterogeneous population.

As we mentioned in Section 3.1.3, the drawback in our method is that any number of clusters can be found. This is not as big a problem for real-life data as it is for laboratory data because we do not know the number of clusters, if any, that should be found. We also assume that the data are clusterable which may not always be the case even for behavioural data collected in a laboratory. As HDBSCAN* is inherently able to estimate the number of clusters and identify some data points as noise, further investigation of the method combined with SOM or some other dimension reduction or manifold approximation method could prove to be useful. Semi-supervised methods that only require partial labelling could also be considered in future studies. Future real-life investigations would also benefit from asking the feeling of stress directly to obtain a solid self-reported stress label and to allow for estimating how stress, abnormal feelings and abnormal behaviour patterns relate to each other.

Another drawback in the model is that it does not take feature autocorrelation into account. Because of this defect, SOM does not understand sequential patterns or especially events of recurring sequential patterns. We did some experiments with a recurrent version of SOM but the results were not satisfying and more research is needed to capture this important aspect. SOM also looks at data at a high and general level and is therefore unable to find detailed and short-term patterns.

In a larger scale we are interested in modelling human behaviour in general and investigating what kinds of patterns and routines can be found and to what feelings and emotions do they associate with. To do this the preceding problems must be overcome either by improving the existing method or by finding some novel approach.

References

Adams, P. et al (2014). Towards personal stress informatics: comparing minimally invasive techniques for measuring daily stress in the wild. In *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*, pages 72–79. ICST.

Alberdi, A., Aztiria, A. and Basarab, A. (2016). Towards an automatic early stress recognition system for office environments based on multimodal measurements: a review. *Journal of Biomedical Informatics*, 59:49–75.

Bakker, J., Pechenizkiy, M. and Sidorova, N. (2011). What's your current stress level? Detection of stress patterns from GSR sensor data. In *2011 IEEE 11th International Conference on Data Mining Workshops*, volume 28, pages 573–580. IEEE.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 1st edition.

Campello, R.J.G.B. et al (2015). Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data*, 10:1–51.

Ciman, M. and Wac, K. (2018). Individuals' stress assessment using human-smartphone interaction analysis. *IEEE Transactions on Affective Computing*, 9(1).

Cohen, S., Kamarck, T. and Mermerlstein, R. (1983). A global measure of perceived stress. *Journal of Health and Social Behaviour*, 24:385–396.

Dempster, A., Laird, N. and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1–38.

Duda, R.O., Hart, P.E. and Stork, D.G. (2000). *Pattern Classification*. Wiley-Interscience, New York, 2nd edition.

Ervasti, M. et al (2019). Exploratory study of mobile stress management app use interest: influence of personality and differences in stress processing among Finnish students. *Journal of Medical Internet Research, Mental Health*, 6(3). Accepted for publication.

EU-OSHA (2013). *Campaign Guide: Managing stress and psychosocial risks at work*. European Agency for Safety and Health at Work.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874.

Gnambs, T. and Staufenbiel, T. (2018). The structure of the General Health Questionnaire (GHQ-12): two meta-analytic factor analyses. *Health Psychology Review*, 12:179–194.

Hakanen, J. (2009). *Työn imun arviointimenetelmä (Utrecht Work Engagement Scale)*. Työterveyslaitos.

Hassard, J. et al (2014). *Calculating the cost of work-related stress and psychosocial risks*. European Agency for Safety and Health at Work.

Haykin, S. (2008). *Neural Networks and Learning Machines*. Pearson Education, Inc., New Jersey, 3rd edition.

Healey, J. (2000). *Wearable and automotive systems for affect recognition from physiology*. Ph.d. dissertation, Massachusetts Institute of Technology.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Huysmans, D. et al (2018). Unsupervised learning for mental stress detection - exploration of Self-Organizing Maps. In *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies*, pages 26–35. SCITEPRESS - Science and Technology Publications.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer Series in Information Sciences. Springer Berlin Heidelberg, Berlin, Heidelberg, 3rd edition.

Kohonen, T.K. (2014). *MATLAB implementations and applications of the Self-Organizing Map*. Unigrafia Oy, Helsinki, Finland.

Kusserow, M., Amft, O. and Troster, G. (2013). Monitoring stress arousal in the wild. *IEEE Pervasive Computing*, 12:28–37.

Lazarus, R.S. (1993). From psychological stress to the emotions: A history of changing outlooks. *Annual Review of Psychology*, 44:1–22.

McInnes, L. and Healy, J. (2017). Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE.

McInnes, L., Healy, J. and Astels, S. (2017). hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2:205.

Pedregosa, F. et al (2011). Scikit-learn : machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pietilä, J. et al (2017). Evaluation of the accuracy and reliability for photoplethysmography based heart rate and beat-to-beat detection during daily activities. In Eskola, H. et al, editors, *EMBEC & NBC 2017*, volume 65 of *IFMBE Proceedings*, pages 145–148. Springer Singapore.

Rousseeuw, P.J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Sano, A. et al (2018). Identifying objective physiological markers and modifiable behaviors for self-reported stress and mental health status using wearable sensors and mobile phones: observational study. *Journal of Medical Internet Research*, 20.

Schmidt, P. et al (2018). Introducing WESAD, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 2018 on International Conference on Multimodal Interaction - ICMI '18*, pages 400–408, New York, New York, USA. ACM Press.

Selye, H.M. (1956). *The Stress of Life*. McGraw-Hill, New York, New York, USA.

Shaffer, F. and Ginsberg, J.P. (2017). An overview of heart rate variability metrics and norms. *Frontiers in Public Health*, 5:1–17.

Sharma, N. and Gedeon, T. (2012). Objective measures, sensors and computational techniques for stress recognition and classification: a survey. *Computer Methods and Programs in Biomedicine*, 108:1287–1301.

Shi, Y. et al (2010). Personalized stress detection from physiological measurements. In *International Symposium on Quality of Life Technology*.

Smets, E. et al (2016). Comparison of machine learning techniques for psychophysiological stress detection. In Serino, S. et al, editors, *Pervasive Computing Paradigms for Mental Health. MindCare 2015*, volume 604 of *Communications in Computer and Information Science*, pages 13–22. Springe, Cham.

Smets, E. et al (2018). Large-scale wearable data reveal digital phenotypes for daily-life stress detection. *npj Digital Medicine*, 1:1–10.

Soto, C.J. and John, O.P. (2017). The next Big Five Inventory (BFI-2): Developing and assessing a hierarchical model with 15 facets to enhance bandwidth, fidelity, and predictive power. *Journal of Personality and Social Psychology*, 113:117–143.

Stefanovič, P. and Kurasova, O. (2011). Visual analysis of self-organizing maps. *Nonlinear Analysis: Modelling and Control*, 16:488–504.

Taylor, S.A. et al (2017). Personalized multitask learning for predicting tomorrow's mood, stress, and health. *IEEE Transactions on Affective Computing*.

Ultsch, A. and Siemon, H. (1990). Kohonen's self organizing feature maps for exploratory data analysis. In *Proceedings of International Neural Networks Conference (INNC)*, pages 305–308.

Vesanto, J. and Alhoniemi, E. (2000). Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, 11:586–600.

Vesanto, J. et al (2000). SOM toolbox for Matlab 5. Technical report, Helsinki University of Technology.

Vildjiounaite, E. et al (2018). Unobtrusive stress detection on the basis of smartphone usage data. *Personal and Ubiquitous Computing*, 22:671–688.

Vildjiounaite, E. et al (2017). Unsupervised stress detection algorithm and experiments with real life data. In Oliveira, E. et al, editors, *Progress in Artificial Intelligence*, Lecture Notes in Computer Science, vol 10423. Springer, Cham.

Wittek, P. et al (2017). somoclu : An efficient parallel library for Self-Organizing Maps. *Journal of Statistical Software*, 78:1–21.

Xu, Q., Nwe, T.L. and Guan, C. (2015). Cluster-based analysis for personalized stress evaluation using physiological signals. *IEEE Journal of Biomedical and Health Informatics*, 19:275–281.